CA SOLVE:Access™ Session Management

Reference Guide

r5

This documentation and any related computer software help programs (hereinafter referred to as the "Documentation") are for your informational purposes only and are subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be used or disclosed by you except as may be permitted in a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Auditor
- CA Mainframe Software Manager (CA MSM)
- CA NetMaster® Network Management for SNA (CA NetMaster NM for SNA)
- CA NetMaster® Network Management for TCP/IP (CA NetMaster NM for TCP/IP)
- CA NetSpy™ Network Performance (CA NetSpy)
- CA SOLVE:Access® Session Management (CA SOLVE:Access)
- CA SOLVE:FTS

Note: This guide contains descriptive text and procedures about options and products that you may not be licensed for or have not enabled. Inclusion of the descriptions of these options and products in this guide in no way implies that you are licensed for these options or products.

Contact CA Technologies

Contact Technical Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Introduction

This section contains the following topics:

Intended Audience

This guide is intended for technical personnel responsible for the planning and maintenance of your product's functions and services. It contains information about the advanced functionality of your product.

Typographic Conventions

This section explains the conventions used when referring to various types of commands and when indicating field attributes.

Convention	Description
Commands	Commands such as SYSPARM and SHUTDOWN are shown in upper case.
User Entries	Information to enter onto panels is displayed in **bold** text.
Cross-References	Cross-reference links to other sections of the book are displayed as underlined blue text.
Shortcuts	Shortcuts to menus or options are displayed in **bold**, for example, **/PARMS**.

Chapter 2: Advanced Customization Tasks

This section contains the following topics:

Customize OCS Function Key Settings

There is support for up to 24 function keys in Operator Console Services (OCS) mode. You can customize these keys either locally or globally.

List Function Key Settings

Use the PF LIST command to list the current settings of the function keys.

Set Local Function Keys

You can customize the function keys of each OCS panel with the PF command. Function keys set from an OCS panel:

- Apply to the requesting panel only
- Are valid only until you exit from OCS

Example

To define F4 as a conversational function key, issue the following command:

```
PF4 CONV,MSG USER1
```

Set Global Function Keys

You can use the PF command to set the default or global values of the function keys that apply on initial entry to OCS. Function keys set in this way apply to all OCS panels unless overridden by a further PF command entered from OCS mode.

To do this, execute the PF command in the BSYS environment (for example, in your INIT or READY procedures) or use the SUBMIT command to pass a PF command to the BSYS environment.

Note: For information about the PF command, see the online help.

Default OCS Function Key Settings

If no PF commands are included in your INIT procedure, the following defaults apply:

Key	Command Issued	Type
F1/13	-FSPROC ##HELP	Prefix
F2/14	SPLIT	Immediate
F3/15	X	Immediate
F4/16	RETURN	Immediate
F5/17	-FSPROC $CMDENT+	Prefix
F6/18	AUTOHOLD	Immediate
F7/19	-FSPROC $LOBROW	Immediate
F8/20	CLEAR	Immediate
F9/21	SWAP	Immediate
F10/22	CS+	Immediate
F11/23	CS-	Immediate
F12/24	ORDER	Immediate

Equate Command Strings

The EQUATES parameter group lets you substitute a long command string with a single one- to eight-character string. The string is then easier and faster to use. This is particularly useful when you are using Remote Operator Facility (ROF). You can also use the EQUATES parameter group to replace an existing command with a Network Control Language (NCL) procedure.

You can set up equates in the EQUATES parameter group when you first customize your region. These equates are then immediately available to any operator using OCS mode. These equates are known as global EQUATES.

Individual users can override a global equate by using the EQUATE command in OCS. Equate values set by an individual user are effective only for that user, and only for the single OCS session.

Note: For more information about the EQUATE command, see the online help.

Replace Commands After Exit Initialization

To substitute commands with NCL procedures after initialization, you can use either the CMDREPLS or EQUATES parameter group.

Note: Command replacement using the EQUATES parameter group takes effect during system initialization only.

Implement the Time-out Facility

You can set terminals that are logged onto an application to time out after a period of inactivity. This reduces the security risk of having them logged on but unattended.

A general time-out facility is provided in your product region. Use this to specify time-out intervals and actions for all terminals.

The time-out facility identifies a user at a terminal as having been inactive for a period and executes an action. Possible actions are to ring the terminal alarm, or to log the user off the system.

When time-out management is implemented, it affects all of the users of your region, unless their user ID definitions exempt them.

The TIMEOUT parameter group lets you administer the time-out facility. You can access parameter groups through the /PARMS panel shortcut.

Customize National Language Character Set Support

Your product supports the use of National Language (NL) character sets. This support is automatic and is controlled by the SYSPARMS LANG command. The default is US; however, many different languages are supported.

Note: For more information about supported language codes and their associated code pages, see the *Network Control Language Reference Guide.*

You can set an individual language code for users in their user ID definitions.

Note: For information about how to specify a language code in a user ID, see the *Security Guide*.

What National Language Support Affects

The following functions perform language support translation using the code page associated with a user's language code:

- NCL panel processing using #FLD CAPS=YES
- NCL statements using the &TRANS built-in function with the NLUPPER and NLLOWER operands

Determine a User's Language Code

The &ZUSERSLC system variable returns the system recognized language code for a user. This is one of the following values:

- The user's language code
- The system language code if the user's language code is not valid
- The value UK if the system language code is not a supported code

Provide a Customized Primary Menu

If you want to customize the primary menu for your installation, you must understand how the supplied primary menu procedure works.

Note: If you want to customize the supplied procedure, ensure that you rigorously test any changes for all user classes before implementing the new procedure.

When you create a new primary menu procedure, specify the name of the new procedure in the PMENUCONTROL parameter group. You can also specify the name using the SYSPARMS MENUPROC command and include it in your INIT procedure.

Supplied Primary Menu Procedure

The primary menu procedure provides an entry point for every user and provides an appropriate selection list of functions for the users of your system.

The name of the supplied primary menu procedure is $NMPMENU. This procedure is invoked by the system to perform system checks and then invokes the procedure, $NMPEXIT, to perform the presentation of the menu.

Both $NMPMENU and $NMPEXIT are standard NCL procedures and are invoked according to each user's NCL library specification.

The primary menu procedure is invoked under the following conditions:

- After initial logon
- At the termination of an OCS or MAI session
- If the procedure terminated without specifying a mode change or window termination

Expired Password Procedure

$NMPMENU must be able to detect when a user's password has expired. The &ZPWSTAT system variable is used for this purpose. The variable invokes the $UAPWD01 procedure to prompt for password change. Make sure that this facility is available if you customize the supplied procedure.

Reconnection Menu

$NMPMENU supplies a reconnection menu, providing multiple-signon users with the following choices of reconnection options after a session outage:

- Reconnect to a particular region
- Bypass reconnection and display the primary menu
- Bypass reconnection and cancel all disconnected regions

Display the Primary Menu

$NMPEXIT is invoked by $NMPMENU to display the primary menu. The menu panel has the following features:

- Fixed title line
- Input option line
- Error message line
- Floating trailer containing the menu broadcast and available function keys
- 55 lines available for menu selections
- A user information box on the right hand side

A maximum of 12 menu options can be displayed on the menu at one time. The format and number of these selections should be displayable on all devices supported by your installation's network.

$NMPEXIT uses the security query capability of &SECCALL to determine users' authorized privileges. It then matches this profile to determine which features they are authorized to access, and displays only these features on the menu.

Alter the Primary Environment

When the primary menu is displayed, the primary environment is operating in base NCL mode. There are two other modes of primary environment:

- OCS
- MAI-FS

These modes are invoked when the primary menu procedure issues a SETMODE command to modify the primary environment. When these alternate modes are terminated, the primary environment returns to base NCL mode.

For example, if a user is authorized to access OCS and enters the O option on the primary menu, OCS is invoked using the SETMODE OCS command. This alters the primary environment so that it operates in OCS mode. When users exit OCS, they are returned to the primary menu, and the primary environment operates in base NCL mode.

Support Single Option Users

If a user is only authorized for one option on the primary menu, the primary menu procedure can detect this situation and will automatically select that option and display the menu for that option. For example, if a user is only authorized for OCS, the primary menu procedure will bypass the primary menu and change to OCS mode automatically. When the user exits the option, the session is closed without displaying the primary menu.

Chapter 3: Initializing Your Region Using the SOLVE Program

This section contains the following topics:

About SOLVE

SOLVE is a program that lets you do the following:

- Circumvent the 100-character JCL parameter limit.
- Start a region before JES without cataloging data sets in the master catalog.
- Use the MVS system symbol substitution service in a control file to tailor program parameters. This aids in cloning systems in a sysplex environment.

The SOLVE program is supplied in object code and has the following product-dependent aliases:

- CONNECTR
- SOLVE

How SOLVE Processing Works

The SOLVE program processes in the following sequence:

1. The NMDRVCTL data set (the control file) opens.
2. Input commands are read and processed.
3. The NMDRVCTL data set closes.
4. If no errors are detected, control is transferred to the target program (using XCTL), passing the input JCL PARM as prefixed, suffixed, or both.

If errors are detected in the input commands or if a dynamic allocation error occurs, the error message is written to the console as a WTO message and the action specified by the ERROR command is taken. The target program is not started in this case.

Specify the NMDRVCTL Data Set

When SOLVE executes, it opens and reads the data set associated with the NMDRVCTL ddname to obtain parameters. This section describes how to specify the contents of NMDRVCTL.

The NMDRVCTL DD statement points to the data set that contains control statements for SOLVE. These control statements specify:

- SOLVE options
- PARM information for the target program
- Data set or SYSOUT allocation requests

The data set must have the following attributes:

- Fixed (blocked or unblocked)
- LRECL 80
- Any valid block size

The data set can be sequential or a partitioned data set (PDS) member, or a concatenation of these. For example, it can be a member in SYS1.PARMLIB.

The control statements must be in the following format:

- All control statements must be in uppercase.
- Only columns 1 through 72 are examined. If line numbers are desired, they can be in columns 73 through 80.
- Blank lines are ignored.
- Any line with an asterisk (*) as the first non-blank character is treated as a comment and ignored.
- Other lines must contain valid SOLVE commands (see the following section for details of these commands).

  Only one command is allowed per line. Commas can separate the operands of a command. Comments can trail on the lines if they are separated from the commands and operands by a blank.
- The commands for data set and SYSOUT allocation can span several lines. In this case, the plus sign can be used as a parameter to indicate that the next line is a continuation of the current command.

SOLVE Commands

The SOLVE program has three groups of commands:

- General
- The dynamic allocation of data sets
- The dynamic allocation of SYSOUT data sets

General Commands

The following general commands can be specified in the NMDRVCTL data set:

PGM={ NM001 | *name* }

Specifies the target program to which control is transferred.

Default: NM001

ERROR={ *tnnnn* | R100 }

Specifies the action to take if an error is detected in the input control statements. The following actions are valid:

R*nnnn*

Returns to the system with *nnnn* return code.

U*nnnn*

Abends with user ABEND code *nnnn*.

Default: R100

PARMSEP=*c*

Specifies the separator character for concatenation of PARM information. The nominated character is used only between the PARM prefix (see the PPREF command), the JCL PARM, and the PARM suffix (see the PSUFF command). The character is not used between individual sections of the prefix and/or suffix.

Default: comma (,)

PPREF=*value*

Specifies the prefix of the supplied JCL parameters that is passed to the target program. This command must not span lines, but can be specified several times. The multiple specifications are concatenated together, in appearance order, with no intervening separators, and treated as a single prefix.

Limits: 1024 characters

PSUFF=*value*

Specifies the suffix of the supplied JCL parameters that is passed to the target program. This command must not span lines, but can be specified several times. The multiple specifications are concatenated together, in appearance order, with no intervening separators, and treated as a single suffix.

Limits: 1024 characters

LIST={ NO | YES }

Controls the listing of input lines. If LIST=YES is specified, all following lines are listed to the console using WTO messages. If SUBS=YES is in effect, each line is also displayed after substitution processing.

SUBS={ NO | YES }

Controls symbolic substitution for line processing. Specifying SUBS=YES enables substitution on all following input lines (except comments).

Specifying SUBS=NO stops substitution on all following input lines.

VAR*xxx*=*'value'*

Lets you define up to 20 additional user variables for substitution. *xxx* is the variable name, and the value is specified after the equal sign (=), which can be quoted if containing blanks. Each variable name can have up to 16 characters.

If substitution is in effect, the variable name and value can be built from other variables.

Note: The value length cannot exceed the length of the actual variable name + 1 (for an ampersand). The symbol substitution service enforces this rule to prevent buffer overruns.

WAIT={ ESM | VTAM }

Specifies whether initialization waits for the External Security Manager (ESM) or VTAM to become available.

Dynamically Allocate a Data Set

The following command and its operands can be used to dynamically allocate a data set:

```
DD=name | DD=*
        [ ,DSN=datasetname ]
        [ ,DISP=( { OLD | SHR | MOD | NEW } [ ,ndisp ]
        [ ,cdisp ] ] ) ]
        [ ,VOL=volume ]
        [ ,UNIT=unitname ]
        [ ,{ CYL=( pri,sec ) | TRK=( pri,sec ) } ]
        [ ,BLKSIZE=nnnnn ]
```

If the operands cannot fit on one line, you can use the plus sign (+) to indicate that the command is continued. For example:

```
DD=FRED,+
        DSN=FRED.DATASET,+
        DISP=OLD
```

The operands are:

DD=*name* | DD=*

Specifies the DD name being allocated. This operand must be first and a valid DD name must be used. DD=* indicates that this data set is to be dynamically concatenated to the previous data set allocation (which must be a data set, not a SYSOUT allocation).

DSN=*datasetname*

Specifies the data set to be allocated. A PDS member name or GDG relative generation number can be specified in parenthesis after the data set name.

DISP=({ OLD | SHR | MOD | NEW } [*,ndisp* [*,cdisp*]])

Specifies the data set disposition. *ndisp* and *cdisp* are the standard UNCATLG, CATLG, DELETE, and KEEP options and default as in standard JCL. If only the first disposition is required, no parentheses are necessary. For example:

```
DISP=OLD
```

VOL=*volume*

Specifies the volume that the data set is (to be) on. For existing data sets, bypasses a catalog search. Generally requires that the UNIT operand also be specified.

UNIT=*unitname*

Specifies the unit type that the data set is (to be) on. This is required for existing data sets if the volume is specified.

CYL=(*pri*,*sec*) | TRK=(*pri*,*sec*)

For new data sets, allows specification of a space allocation value.

Note: You cannot allocate a PDS because no directory amount can be specified.

BLKSIZE=*nnnnn*

Lets you specify a block size. This is also useful when concatenating data sets of unlike block size because it sets the DCB BUFLEN value.

Dynamically Allocate a SYSOUT Data Set

The following command and operands may be used to dynamically allocate a SYSOUT data set:

```
DD=name
          SYSOUT=class
         [FREE={ CLOSE | END } ]
         [HOLD={ NO | YES } ]
         [BLKSIZE=nnnnn ]
```

If the operands will not fit on one line, an operand, you can use the plus sign (+) to indicate that the command is continued. For example:

```
DD=LOG1,+
         SYSOUT=A,+
         FREE=END
```

The operands to dynamically allocate a SYSOUT data set are:

DD=*name*

Specifies the DD name being allocated. Must be the first operand. A valid DD name must be used.

SYSOUT=*class*

Specifies the SYSOUT class desired. Use a single letter or number, A to Z, or 0 to 9. Specifying an asterisk implies that the job MSGCLASS is to be used (as in JCL).

FREE={ CLOSE | END }

CLOSE specifies that the SYSOUT is to be spun-off when closed, END specifies that it is not. Note that FREE=CLOSE is the default. You *should not* use FREE=CLOSE for FMTDUMP.

HOLD={ NO | YES }

Specifies if the SYSOUT is to be held or not. HOLD=NO is the default.

BLKSIZE=*nnnnn*

Lets you specify a block size.

Chapter 4: Tuning Performance

This section contains the following topics:

Enhance Performance

You can enhance the performance of your regions at each of the following levels:

- System
- Product
- Function

By following the performance enhancement measures outlined in this chapter, you will achieve the most effective use of your product.

Tune at the System Level

The first area to consider when tuning your region is at the system level. The following points outline system level performance considerations:

- For z/OS systems, the performance group and dispatching priority can affect the performance of your region. High-priority systems should be in the same domain as VTAM and just below it in dispatching priority.
- In terms of tuning, your product can be regarded as equivalent to CICS or IMS. All system tuning rules that apply to other online systems should be used with these systems to ensure that your product region is not starved for real storage.
- Data set placement is important to consider when increasing the performance of your system. The PANELS, MODS, and NCL COMMANDS data sets can have a large amount of I/O activity on them. For this reason, consider placing these data sets on volumes and paths that are not otherwise busy.
- Running your product region as non-swappable can help to increase response time, which is important when the overall activity rate is low but quick response is important. The NONSWAP parameter group is used to specify whether your product region is non-swappable (the default).

Note: Some products must run non-swappable.

Tune VTAM Interface

To help ensure that you gain maximum effectiveness from your products, consider the following performance points for your VTAM interface:

- Review any pacing specifications in System Services APPL statement parameters. They could affect INMC performance.
- If INMC is implemented using VTAM, dedicated COS definitions can be specified to optimize paths through the network (for multi-region environments only).

Tune Panel Use

In some products, a large number of panels may need to be displayed. These panels are stored on VSAM data sets. Panels are retrieved from these data sets as required. Panel definitions are retained in storage for both reuse and automatic sharing.

Tune Storage Limits for Panel Sends

The maximum amount of storage allocated to allow panels to be sent to terminals is crucial to the performance of your region and is separate to considerations for tuning panel access. The following two SYSPARMS operands are used to specify storage amounts:

PANLBFSZ

Sets the amount of storage to be acquired to build a panel data stream. The default is 20 KB.

PANLBUFF

Sets the maximum number of pages of storage to be allowed for panel buffers

To avoid severe response time problems when you have a large number of application users, increase the value of the PANLBUFF operand. Otherwise, when a user requests a panel, they might have to wait until a previous panel send has been completed before they have access to their panel.

More information:

SYSPARMS Operands (see page 131)

Monitor Panel Send Storage Limits

To monitor how your system performs panel sends, use the SHOW GRP=PBUFPGT command. This command shows statistics about the panel buffer storage pool and can help you determine whether delays are being caused by panel send throttling.

The reason that the current storage utilization may be close to, or above, the defined maximum storage, is that some of the terminals that have had panels sent to them have not returned a definite response. This causes unnecessary throttling of panel send operations.

The system automatically increases the limit temporarily if the condition persists and returns the limit to its original value when the condition is relieved. This does not alter the short-term throttling characteristic, which is used to prevent flooding of the network and VTAM.

Note: You can alter the throttling characteristics dynamically by issuing the SYSPARMS PANLBUFF command to increase or decrease the active panel send limit. For example, to minimize the effect of panel sends not completing, you could increase the panel send limit after EASINET startup.

Improve NCL Procedure Usage

With some products or user-written applications, a concern may be the rate at which NCL procedures are loaded into storage. Modular programming practices encourage breaking an application into as many separate small modules as possible. This leads to many loading requests for procedures issued as NCL executes.

If this is not addressed, the following problems can occur:

- Applications run slowly, as they are constantly waiting for procedures to load. This is particularly evident in some applications that call a procedure to edit each field on a panel.
- I/O to the NCL procedure libraries can be a bottleneck.
- CPU time is expended in loading and pre-compiling the NCL.
- Real storage is tied up for I/O buffers.

Monitor NCL Procedure Loading Activity

To monitor the loading of NCL procedures for your system, use the SHOW NCLSTAT command. This command displays statistics about the number of:

- Load requests
- Actual loads
- Loads satisfied by preload
- Loads satisfied by autoshare and retain

By monitoring the increase in the value of these statistics across the day, you can use the following techniques to help strike a balance between loading activity and storage use:

- Sharing NCL procedures between users
- Preloading NCL procedures

Preload NCL Procedures

You can nominate a set of NCL procedures to remain preloaded by using the OCS command, LOAD. This involves a once-only load and precompilation. These procedures can then be shared by any number of users. Preloading NCL procedures is useful when you have short routines that are typically used by a single user (no concurrent usage). These procedures are constantly flushed from storage and reloaded again. It is also best to specify any heavily-used NCL procedures to be preloaded.

Note: For information about the LOAD command, see the online help.

Tune VSAM Data Sets

Most of these products rely on VSAM UDBs and NDBs. The way you manage your VSAM data sets has a direct effect on the performance of your products. The facilities described in the following sections allow you to tune UDBs and NDBs for maximum performance benefits.

Buffer Sharing

Buffer sharing is controlled by the VSAM LSR (Local Shared Resources) facility. By sharing buffers, data set I/O is reduced. The shared buffers are defined in LSR in pools.

To enable buffer sharing for a VSAM data set, use the UDBCTL OPEN command with the LSR operand.

Note: For files allocated through Customizer parameter groups, you can enter UDBCTL operands as VSAM options. For more information about the UDBCTL command, see the online help.

When specifying the LSR pool definition that best suits your needs, consider the following factors:

- Your need for virtual storage versus I/O performance
- The mix of VSAM files that you have open concurrently in the address space and which of those you want to place in the LSR pool

To define your LSR pool definitions, use the LSRPOOL parameter group (enter **/PARMS**).

Note: VSAM alerts warn about string or buffer shortages.

Deferred Write Capabilities

Deferred write capabilities let you defer the updating of data sets so there are fewer I/O requests. The advantages of using this facility should be weighed against the possibility of losing data in the event of a system failure.

To allow a data set to have deferred write capabilities, specify the DEFER operand on the UDBCTL OPEN command, or VSAM options in a Customizer parameter group.

For more information about the UDBCTL command, see the online help.

VSAM Processing in a Subtask

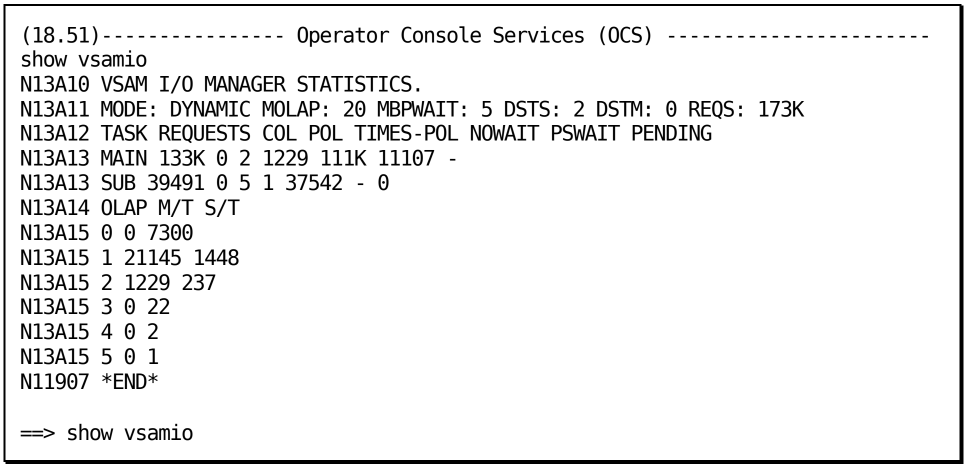
You can nominate VSAM I/O to be performed by a subtask rather than the System Services main task. This provides increased throughput by overlapping VSAM processing. To nominate a subtask to perform VSAM I/O, use the VSAMIO JCL parameter.

Note: Using a subtask is only useful when there is significant VSAM processing and you have a multi-CPU machine.

You can use the SHOW VSAMIO command to obtain statistics of VSAM processing.

Example

Following is an example of SHOW VSAMIO command output:



```
(18.51)---------------- Operator Console Services (OCS) -----------------------
show vsamio
N13A10 VSAM I/O MANAGER STATISTICS.
N13A11 MODE: DYNAMIC MOLAP: 20 MBPWAIT: 5 DSTS: 2 DSTM: 0 REQS: 173K
N13A12 TASK REQUESTS COL POL TIMES-POL NOWAIT PSWAIT PENDING
N13A13 MAIN 133K 0 2 1229 111K 11107 -
N13A13 SUB 39491 0 5 1 37542 - 0
N13A14 OLAP M/T S/T
N13A15 0 0 7300
N13A15 1 21145 1448
N13A15 2 1229 237
N13A15 3 0 22
N13A15 4 0 2
N13A15 5 0 1
N11907 *END*

==> show vsamio
```

More information:

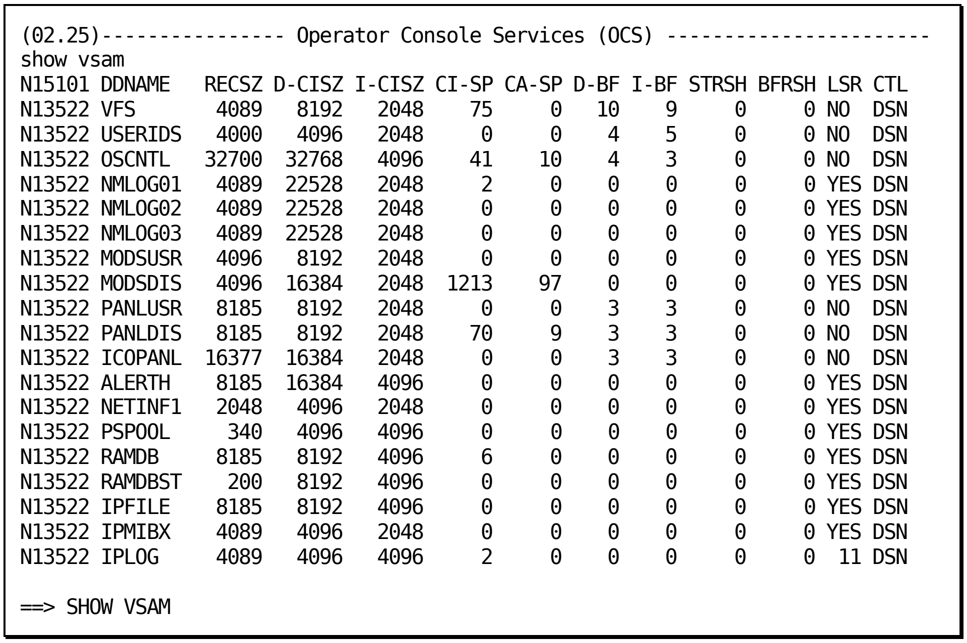
Product Region JCL Parameters (see page 121)

Performance Monitoring Facilities

The SHOW VSAM command is provided to display attributes and statistics about VSAM databases.

Example

Following is an example of a SHOW VSAM display.



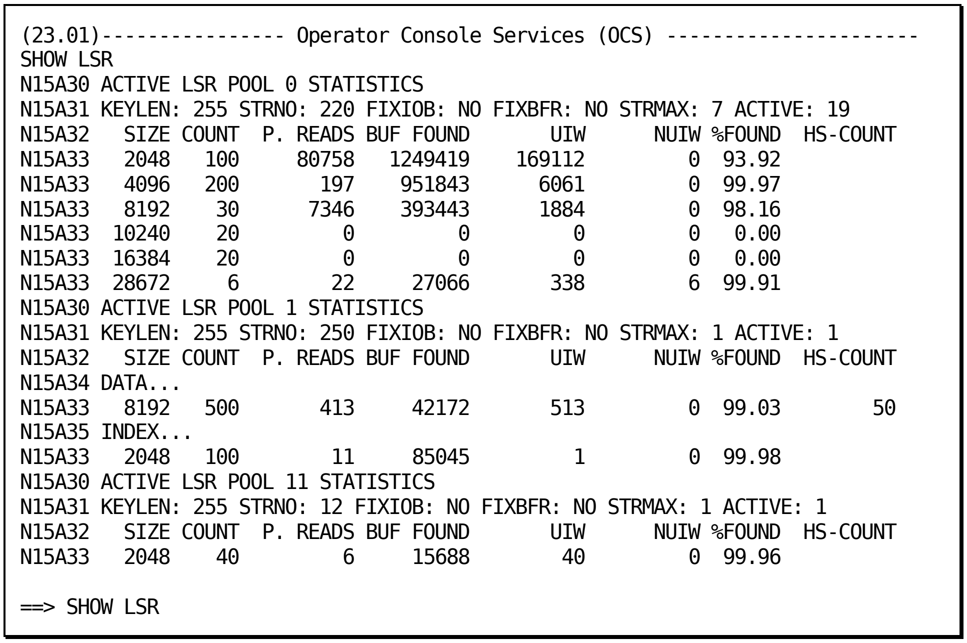
```
(02.25)---------------- Operator Console Services (OCS) -----------------------
show vsam
N15101 DDNAME   RECSZ D-CISZ I-CISZ CI-SP CA-SP D-BF I-BF STRSH BFRSH LSR CTL
N13522 VFS       4089   8192   2048    75     0   10    9     0     0 NO  DSN
N13522 USERIDS   4000   4096   2048     0     0    4    5     0     0 NO  DSN
N13522 OSCNTL   32700  32768   4096    41    10    4    3     0     0 NO  DSN
N13522 NMLOG01   4089  22528   2048     2     0    0    0     0     0 YES DSN
N13522 NMLOG02   4089  22528   2048     0     0    0    0     0     0 YES DSN
N13522 NMLOG03   4089  22528   2048     0     0    0    0     0     0 YES DSN
N13522 MODSUSR   4096   8192   2048     0     0    0    0     0     0 YES DSN
N13522 MODSDIS   4096  16384   2048  1213    97    0    0     0     0 YES DSN
N13522 PANLUSR   8185   8192   2048     0     0    3    3     0     0 NO  DSN
N13522 PANLDIS   8185   8192   2048    70     9    3    3     0     0 NO  DSN
N13522 ICOPANL  16377  16384   2048     0     0    3    3     0     0 NO  DSN
N13522 ALERTH    8185  16384   4096     0     0    0    0     0     0 YES DSN
N13522 NETINF1   2048   4096   2048     0     0    0    0     0     0 YES DSN
N13522 PSPOOL     340   4096   4096     0     0    0    0     0     0 YES DSN
N13522 RAMDB     8185   8192   4096     6     0    0    0     0     0 YES DSN
N13522 RAMDBST    200   8192   4096     0     0    0    0     0     0 YES DSN
N13522 IPFILE    8185   8192   4096     0     0    0    0     0     0 YES DSN
N13522 IPMIBX    4089   4096   2048     0     0    0    0     0     0 YES DSN
N13522 IPLOG     4089   4096   4096     2     0    0    0     0     0  11 DSN

==> SHOW VSAM
```

Monitor LSR Tuning

To assist with specifying the LSR pool definitions that provide the best performance for your system, use the SHOW LSR command. This command displays statistics about the LSR pool. From this display you can note trends and alter the LSR pool definition appropriately.

Example

The following panel displays LSR pool statistics from a system that has been customized, and as a result has significantly more buffers than the distributed definition.



```
(23.01)---------------- Operator Console Services (OCS) ----------------------
SHOW LSR
N15A30 ACTIVE LSR POOL 0 STATISTICS
N15A31 KEYLEN: 255 STRNO: 220 FIXIOB: NO FIXBFR: NO STRMAX: 7 ACTIVE: 19
N15A32   SIZE COUNT  P. READS BUF FOUND       UIW      NUIW %FOUND  HS-COUNT
N15A33   2048   100     80758   1249419    169112         0  93.92
N15A33   4096   200       197    951843      6061         0  99.97
N15A33   8192    30      7346    393443      1884         0  98.16
N15A33  10240    20         0         0         0         0   0.00
N15A33  16384    20         0         0         0         0   0.00
N15A33  28672     6        22     27066       338         6  99.91
N15A30 ACTIVE LSR POOL 1 STATISTICS
N15A31 KEYLEN: 255 STRNO: 250 FIXIOB: NO FIXBFR: NO STRMAX: 1 ACTIVE: 1
N15A32   SIZE COUNT  P. READS BUF FOUND       UIW      NUIW %FOUND  HS-COUNT
N15A34 DATA...
N15A33   8192   500       413     42172       513         0  99.03        50
N15A35 INDEX...
N15A33   2048   100        11     85045         1         0  99.98
N15A30 ACTIVE LSR POOL 11 STATISTICS
N15A31 KEYLEN: 255 STRNO: 12 FIXIOB: NO FIXBFR: NO STRMAX: 1 ACTIVE: 1
N15A32   SIZE COUNT  P. READS BUF FOUND       UIW      NUIW %FOUND  HS-COUNT
N15A33   2048    40         6     15688        40         0  99.96

==> SHOW LSR
```

In this example, little use is being made of the 512-byte buffers. The sum of pool reads plus the number of buffers found is a small number. This small number is a sign that there are too many buffers in the pool. Deleting the pool might also be an option but could also be quite wasteful of storage as buffers from the next larger pool will then be used.

There is a low percentage of 12-KB and 16-KB pools found. This is an indication that the number of buffers could be increased for those pool sizes.

Tune INMC

There are two areas in which INMC tuning can be achieved:

- INMC buffer size—specified by the SYSPARMS INMCBFSZ operand
- BIND RUSIZE—specified in the logmode table definition

If you increase the INMCBFSZ value and the RUSIZE value is not at least this large, no benefits are derived.

INMC also lets logmodes be used with pacing turned on. This feature can prevent a VTAM link being overrun by INMC traffic.

More information:

SYSPARMS Operands (see page 131)

Tune ROF

To prevent excessive message traffic being sent across INMC links during ROF connections, the following techniques can be used:

- Always configure the remote user ID to receive only relevant message traffic.
- Avoid duplication of messages. Operators should only receive unsolicited messages in remote domains if they really need them.
- Consider signing on background regions across a ROF link. For example, if BMON is signed on to a remote domain, any messages it is eligible to receive from the remote domain may be processed by a MSGPROC in the receiving system, and then propagated to all MONITOR receivers in that system. Only one copy of the message flows across the link in this case.

  In this case operators that need to issue commands to the remote domain do not need to be profiled to receive unsolicited message receipt.
- Consider using Inter-System Routing (ISR) facilities if applicable.
- Use the SIGNOFF command to terminate remote connections and free resources.

Control Message Flow in OCS

To ensure that OCS activity impedes the performance of your systems as little as possible, the following points should be considered:

- Restrict the types of messages that an operator receives using UAMS or an external security definition.
- Do not allow all operators to receive unsolicited messages by default.
- Use MSGPROCs. Held message limits can be reached while the panels are being displayed. If messages are not going to be examined, they should not be sent. These messages waste storage space and CPU time.
- Restrict the use of non-roll delete (NRD) messages. Extra system resources are needed to retain NRD messages.
- Encourage the use of selective commands. Monitor the use of $CMDENT. If excessive, perhaps the default wrap count should be lowered.
- Use the SYSPARMS HELDMSG=*xxx*,*yyy* command to limit the hold message queue. The queuing of messages causes excessive storage usage and fragmentation.

Performance Considerations When Writing NCL

When writing NCL, the following points should be considered to ensure that your system performance does not suffer:

- Ensure that the &CONTROL RESCAN option is only used around statements to which it applies. Rescanning every NCL statement slows the system down.
- Use suppressed comments because they occupy no storage. They add an insignificant overhead to the procedure load process only.
- Use modular programming.
- Use the NCLTEST command when testing NCL to circumvent the retain and autoshare facility.
- Do not use static and dynamic PREPARSE facilities for panels unless they are essential to a specific panel display.
- In any NCL procedures that display panels, think about the number of variables that are active across any panel displays. The large number of concurrent processes executing can result in excessive virtual storage usage. Comment out any unneeded variables before any of these displays.
- Reduce the overheads of using &CALL repeatedly to call a module by using one of the following methods:
  - Use the LOAD MOD=*module_name* command, if the module is reentrant, to load it once into storage.
  - Use the SUBSYS facility to attach the program once.

More information:

Improve NCL Procedure Usage (see page 32)

Use Commands to Control Resource Consumption by NCL

There are several commands that can be used to alter the performance of your region. These system performance commands include SYSPGT (see page 48), SYSRCT (see page 49), SHOW SYSWAIT (see page 46), SHOW SYSPGT (see page 41), and SHOW SYSRCT (see page 43). These commands display and control resource consumption by NCL processes.

All NCL statements are assigned a processing unit weighting. SYSRCT and SYSPGT commands let you set up performance groups whereby an NCL process is placed into a forced wait if it consumes a certain amount of these processing units. This allows System Services to run with a high priority in the operating system, but prevents them from using all available CPU time if an NCL procedure starts looping.

System-level procedures should have a high priority. Short-term performance controls can be used to stop a runaway system procedure without impacting it in the long term.

The SHOW NCL command can also be used to indicate the performance group, priority, and current processing unit consumption. For more information about the SHOW NCL command, see the online help.

Important! Busy system procedures that do not wait because there is always a message available for &xxxREAD may be unfairly penalized by the short-term performance control. These should be reset using the following command:

```
SYSRCT G=4 P=0 SDELAY=(0,0,0,0)
```

Performance and Tuning Commands

The following commands set the performance parameters of the different categories of process that your region executes. The commands also display the relative activity in a region.

The default settings are designed to cater for most installation requirements, but these commands are available to change the defaults if necessary.

Important! These commands should be used with caution by skilled system administrators.

SHOW SYSPGT

The SHOW SYSPGT command displays the summary information about the NCL system performance group tables.

The system performance group table contains historical statistics about each of the four performance groups. The table also contains the priority that is given to NCL procedures when they start executing under the control of a particular group. The statistics show the number of currently active procedures, the total started and totals of performance control actions taken for each group.

This command has the following format:

```
SHOW SYSPGT
(0)
```

Example

```
SHOW SYSPGT
N58013 PG--IPRTY-ACTV-UPDATED---STARTED---DELAYS----CPRTY
N58004  1    1      3 14.42.05       13        0        0 BACKGROUND
N58004  2    1      2 14.42.05      121       17        3 OCS
N58004  3    2      0 14.42.05        3        0        0 FULLSCREEN
N58004  4    1     57 14.42.05     2288       54       24 SYSTEM
N13503 *END*
```

Return Information

PG

The performance group number.

IPRTY

The priority that is assigned to a procedure when it starts executing.

ACTV

The number of currently active procedures.

UPDATED

The time RCT control values were set. Initially, these values are set during system start-up.

STARTED

The total number of procedures that have run in this performance group.

DELAYS

The total number of forced waits that have occurred for all procedures that have run in this performance group.

CPRTY

The total number of priority changes that have occurred for all procedures that have run in this performance group.

text

The descriptive name of the performance group.

Performance Groups

BACKGROUND

These are procedures that execute without a physical terminal environment: BGMON and BGLOG.

OCS

These are line mode procedures executing in association with a real OCS window, including commands from ROF sessions.

FULLSCREEN

These are procedures which use panel services, including EASINET.

SYSTEM

These are system-level procedures: BSYS, PPOPROC, CNMPROC, LOGPROC, AOMPROC, RMINIT, and RMREADY procedures, and CA SOLVE:FTS commands.

SHOW SYSRCT

This SHOW SYSRCT command displays summary or detail information about NCL performance groups. It can be used to obtain either summary information about all four NCL performance groups, or detailed statistics about a specific NCL performance group.

This command has the following format:

```
SHOW SYSRCT
(0)      [ =SUMMARY | =n  ]
```

=SUMMARY | =*n*

The default (=SUMMARY) displays a single line for each performance group.

If =*n* is specified, detailed information about the nominated performance group is shown, with all priority levels displayed.

Example

```
SHOW SYSRCT=4
N58007 PERFORMANCE GROUP 4, SYSTEM , 1430 STARTED SINCE - 14.42.05
N58008 PRIORITY+--TRIGGERS-+-------CONTROLS-------+--------STATISTICS-------+
N58009   3      DELAY CPRTY INIT. ADJ. LIMIT NPRTY DELAYS  LIMIT  CPRTY   ACT
N58010 SHORT      -     -     0    0     0    3        -      -      -     0
N58011 LONG       -     -     0    0     0    3        -      -      -
N58009   2
N58010 SHORT      -     -     0    0     0    2        -      -      -     0
N58011 LONG       -     -     0    0     0    2        -      -      -
N58009   1 <=I=
N58010 SHORT     20    10    50    0    50    0        0      0      0    54
N58011 LONG       -    80     0    0     0    0        -      -     13
N58009   0
N58010 SHORT     20     -    50    0    50    0        0      0      -     3
N58011 LONG       -    80     0    0     0    1        -      -     10
N13503 *END*
```

Return Information

PG

The performance group number

ACTV

The number of currently active procedures.

INTERVAL

The statistics recording start time.

STARTED

The total number of procedures that have run in this performance group.

DELAYS

The total number of forced waits that have occurred for all procedures that have run in this performance group.

LIMIT

The total number of times that a forced wait has been issued for the maximum or minimum duration.

CPRTY

The total number of priority changes that have occurred for all procedures that have run in this performance group.

Detailed Information

The detailed information returned includes the previous statistics for long-term and short-term performance control measures, and the following performance control settings for priorities and forcing procedures to wait:

DELAY

The number of processing units that can be consumed by a procedure before a forced wait occurs.

CPRTY

The number of processing units that can be consumed by a procedure before a priority change occurs.

INIT

The length of time in hundredths of a second that a procedure is delayed for, after consuming the DELAY number of processing units for the first time.

ADJ

The length of time in hundredths of a second that the initial DELAY is adjusted by for subsequent times that the delay number of processing units is consumed.

LIMIT

The maximum or minimum length of time a procedure is forced to wait.

NPRTY

The new priority that is assigned to a procedure when it has consumed the CPRTY number of processing units.

The Resource Control Table contains controls for dynamically altering the performance of currently active NCL procedures based on their consumption of *processing units* and statistics recording their effects.

Each priority in a performance group has short term and long-term controls. Short-term controls apply to processing units consumed since the last voluntary loss of control, for example, an &PANEL statement. Long-term controls apply to processing units consumed since the procedure was initiated.

The four performance groups are BACKGROUND, OCS, FULLSCREEN, and SYSTEM, numbered 1 through 4 respectively.

Each performance group has an initial priority which can be displayed using the SHOW SYSPGT command. In the detailed display of the resource controls for a particular group (that is, SHOW SYSRCT=*n*), the initial priority is indicated by <=I=.

SHOW SYSWAIT

The SHOW SYSWAIT command displays main task apparent wait statistics.

The system maintains wait statistics each time an operating system WAIT is issued. The information returned by this command includes the following statistics:

- Percentage of elapsed time that your product region is in an operating system wait (AWAIT%)
- Percentage of times a wait occurred when an NCL procedure was forced to wait due to performance controls (FWAIT%).

These statistics can be used as a guide to how busy the system is and whether performance control measures are having an overall effect.

This command has the following format:

```
SHOW SYSWAIT
(0)
```

Example

```
SHOW SYSWAIT
N58001 PERIOD START:   AWAITS   FWAITS   FWAIT%   AWAIT%     RATIO
N58002   14.30.04      135385     3558    2.62%   99.36%         2
N58002   14.30.04      135385     3558    2.62%   99.36%         2
N58020 PERIOD START:    TOTAL-CPU %-BSY
N58021                 SYSTEM-CPU %-SYS  ACCNTED-CPU %-ACC UNACCNTD-CPU %-UNA
N58022   14.30.04      140.173343   0.6
N58023                   3.118633   2.2   137.047259  97.7     0.007450   0.0
N58022   14.30.04      140.173343   0.6
N58023                   3.118633   2.2   137.047259  97.7     0.007450   0.0
N13503 *END*
```

Return Information

START

The start time for statistics accumulation.

AWAITS

The number of operating system waits that have occurred since the start time.

FWAITS

The number of FWAITS that took place when an NCL procedure was in a forced wait due to performance control measures.

FWAIT%

The percentage ratio FWAITS is to AWAITS.

AWAIT%

The percentage of elapsed time that the system was in an operating system wait.

RATIO

The number of interval events processed per AWAIT.

Use SYSWAIT statistics as a guide only because they reflect the voluntary loss of control of your product region and do not reflect operating system resource management activity.

The statistics are maintained on a historical and interval basis. The first line of the two detail lines shows the statistics since this command was last issued. The second detail line shows the long-term accumulation. These statistics are reset at midnight. Each time this command is issued, the current statistics for this interval are rolled over into the historical statistics. This feature provides a means for displaying activity in the short term, compared against the long-term average.

AWAIT% reflects the apparent wait time that control has voluntarily been passed to the operating system. Therefore, the actual wait time could be higher, because of page faults, higher priority work, and so on.

The number of times that at least one NCL process was forced idle when a wait was issued is also shown as the FWAIT count. This count indicates that work could have been done, but performance controls forced all ready NCL processes to wait.

SYSPGT

The SYSPGT command is used to set the initial priority for an NCL performance group. It is used as part of performance control. It sets the priority that is assigned initially to procedures running in a performance group.

This command has the following format:

```
SYSPGT   GROUP=n
         IPRTY=p
(4)
```

GROUP=*n*

This is the number of the performance group that is to have its initial priority set. The valid range is 1 to 4.

IPRTY=*p*

This is the initial priority for the specified performance group. The valid range is 0 to 3, zero being the lowest priority.

Examples

```
SYSPGT G=1 IP=1
SYSPGT GROUP=4 IPRTY=3
SYSPGT GR=2 I=2
```

The default initial priority for all performance groups can be determined using the SHOW SYSPGT command.

The four performance groups are BACKGROUND, OCS, FULLSCREEN and SYSTEM, numbered 1 to 4, respectively.

A procedure can have its priority altered during execution depending on control values set in the Resource Control Table. See the description of the SYSRCT command for details.

SYSRCT

The SYSRCT command is used to set performance control parameters in the system Resource Control Table (RCT). It is used for performance control. It can be used to set *processing unit* trigger values and performance control parameters. The following types of performance control measure are supported:

- A priority change control measure
- A forced wait control measure

Performance is calculated in the following ways:

- Short-term—that is, between voluntary waits, for example, screen interaction. The SDELAY and SNPRTY operands are used for short-term evaluation.
- Long-term—that is, for the *life* of the NCL process. The LDELAY and LNPRTY operands are used for long-term evaluation.

This command has the following format:

```
SYSRCT   GROUP=n
         PRIORITY=p
         [ LDELAY=(  [ trigger ], [ initial ], [ adjustment ], [ limit ] )  ]
         [ SDELAY=(  [ trigger ], [ initial ], [ adjustment ], [ limit ] )  ]
         [ LNPRTY=(  [ trigger ], [ new-priority ] )  ]
         [ SNPRTY=(  [ trigger ], [ new-priority ] )  ]
(4)
```

GROUP=*n*

The number of the performance group (range 1 to 4) that is to have its RCT values updated.

PRIORITY=*p*

The priority (range 0 to 3, zero being the lowest) that is to be updated.

LDELAY=([*trigger*], [*initial*], [*adjustment*], [*limit*])

SDELAY=([*trigger*], [*initial*], [*adjustment*], [*limit*])

These operands are used to set values for forcing procedures to wait. The trigger value specifies the number of 'processing units' that a procedure can consume before a wait is forced. The three parameters initial, adjustment and limit, specify times in hundredths of a second with a maximum absolute value of 200 (that is 2 seconds).

The initial value is used for setting the period of the first wait. Subsequent wait times might be longer or shorter, being incremented or decremented by the adjustment value, which can be negative or zero. Consequently, the limit can be a maximum or minimum wait time. A zero adjustment results in the limit being set to the initial value, overriding any existing or specified value. When no limit value has been set and a positive adjustment has been specified a default maximum limit of 200 is used.

LNPRTY=([*trigger*], [*new-priority*])

SNPRTY=([*trigger*], [*new-priority*])

These operands are used to set values for altering the priorities of procedures as they execute. The trigger value specifies a number of processing units that can be consumed before a priority change occurs. The new priority must be 0 to 3 and can be the same as the current priority.

Examples

```
SYSRCT G=1 P=1 LNP=(100,2)
SYSRCT GROUP=1 PRIORITY=2 SDELAY=(20,5,1,20)
SYSRCT GR=1 PR1=2 SDELAY=(,,,30)
```

The first example sets a long-term priority change for procedures in group 1 (BACKGROUND). Priority 1 is to be altered to priority 2 after the consumption of 100 *processing units*. The SYSPGT command can have been used to set the initial priority for this performance group to be 1.

The second example sets a forced wait for procedures running in priority 2 to occur every 20 *processing units*. The first wait would be for an interval of 5 hundredths of a second, the second would be 6, the third 7, and so on, until the maximum of 20 hundredths (0.2 seconds) is reached.

The third example alters only the short term delay value limit to be 30 hundredths of a second.

Processing units are a simple measure of work arbitrarily assigned to the execution of NCL statements. They are in no way an accurate measure of actual work performed, or an approximation of CPU time consumed. You cannot compare the work done by two different NCL processes based on processing units consumed.

There are two sets of processing unit consumption statistics for each NCL procedure—one for short-term evaluation and the second for long-term evaluation. It also maintains a long-term dispatching priority as well as a current dispatching priority. Short-term statistics are reset whenever a voluntary loss of control occurs or a priority change takes place. A short-term priority change alters only the current dispatching priority, whereas a long term priority change alters both priorities.

The order of evaluation of performance control measures is as follows:

1. Long-term forced wait and priority change check
2. Short-term forced wait and priority change check

Only the first of the above that is triggered is actioned; however, both a priority change and a wait can occur at the same time, with the wait being actioned before the priority change. Procedures started using an &INTCMD statement are run in the owning process's performance group, starting at the performance group's initial priority.

The four performance groups are BACKGROUND, OCS, FULLSCREEN, and SYSTEM, numbered 1 to 4 respectively.

Record CPU Usage

This section describes how to use the CPU-time accounting facilities of NCL.

NCL CPU-time Accounting

NCL CPU-time accounting lets you collect NCL CPU usage data at user level. This data can then be used to generate reports on the CPU usage of each user.

NCL CPU usage data is provided by the generation of System Management Facility (SMF) records at regular time intervals. These records contain the user ID, and the amount of CPU-time used by that user, including multiple signons, and background APPC or ROF regions.

You can control the time interval at which records are generated, and the CPU-time threshold at which a record is generated using the USERACCT (see page 55) command.

You can also display the current status of CPU-time accounting (see page 58).

Implement NCL CPU-time Accounting

To implement NCL CPU-time accounting, the OPT=01 JCL parameter must be specified and SMF record generation must be enabled by the SMF parameter group. Your system must also be running *authorized* for SMF reporting to occur.

Note: CPU-time accounting might impede the performance of your system.

Generation of SMF Records

SMF records are generated under the following circumstances:

- If the amount of CPU-time used by the user since the last record was generated, surpasses a predefined threshold
- On system initialization if user NCL CPU-time accounting is active at the time of shutdown

Whenever an SMF record is generated for a user, their CPU-time is reset to zero. Each SMF record contains the amount of CPU-time used by the user since the previous SMF record was generated.

Adjust Threshold Levels and Report Intervals

You can generate records for all of the users by changing the threshold value to zero. This gives you a cutoff value for CPU usage at the end of the day.

Creating reports at intervals during the day is useful if the system fails. There will still be CPU usage information available.

Setting a minimum threshold level obtains update information for the heavy CPU users. Time is not wasted generating incremental reports for users who use little or no CPU time.

SMF Record Format

The SMF record format consists of the following sections:

- Standard SMF record header
- User ID
- CPU-time

The format is shown in the following table.

Offsets	Length	Format	Source	Description
Standard SMF Header				
0	2	binary	internal	Record length
2	2	binary	internal	Segment descriptor (0000 as record not spanned)
4	1	binary	SVC 83	System indicator
5	1	binary	user supplied	Record type
6	4	binary	SVC 83	The time since midnight in hundredths of a second that record was moved to SMF buffer
10	4	packed	SVC 83	The date the record was moved to the SMF buffer, in the form 00YYDDDF (where F is the sign)
14	4	character	SMCASID	System identification (taken from the SID parameter)
Prefix				
18	1	binary	internal	Sub-category: X’06’ for user CPU-time accounting
19	1	null	-	Reserved

Offsets	Length	Format	Source	Description
20	12	character	internal	NMID of product region
CPU-time Accounting Section				
32	8	character	internal	User ID to which the CPU-time value applies
40	1	binary	internal	Code indicating where CPU-time was used (X’00’ is NCL CPU-time)
41	1	null	-	Reserved
42	8	stck	internal	CPU-time accumulated by user since last SMF write (stck format, where bit 51=1 microsecond)
50	4	binary	SVC 83	The time since midnight, in hundredths of a second, that the last record was cut for this user (0 if first record)
54	4	packed	SVC 83	The date the last record was cut for this user, in the form 00YYDDDF (where F is the sign) (null if first record)

Start and Stop CPU-time Accounting

The USERACCT command is used to start and stop NCL CPU-time accounting. It is also used to set the SMF record ID, and parameters controlling the timing and level of SMF reporting. Some of these parameters affect the time interval and start time of SMF report generation, so they can only be set if NCL CPU-time reporting status is inactive.

USERACCT Command

The USERACCT command has the following format:

```
USERACCT  START
          [ SMFRECID=smf-record-id ]
          [ INTERVAL=hours-between-smf-records
            [ FROM=start-time-for-first-report ]
            [ MINCPU=minimum-CPU-time-to-report ] ]

USERACCT  SET
          [ SMFRECID=smf-record-id ]
          [ INTERVAL=hours-between-smf-records
            [ FROM=start-time-for-first-report ]
            [ MINCPU=minimum-CPU-time-to-report ] ]

USERACCT  STOP
(4)
```

START

Activates NCL CPU-time reporting. This operand controls whether SMF reports on the statistics are generated. The actual accumulation of CPU-time statistics occurs independently of the reporting process, thus CPU-time statistics are collected irrespective of whether reporting is active or inactive.

This operand is accepted only if reporting is currently inactive.

SMFRECID=smf-*record-id*

Specifies the SMF record ID to be used on SMF records that contain NCL accounting information. It is effective immediately after the command is issued, so the next record generated uses this smf-record-id value.

Limits: 128 to 255.

Default: The default SMF record ID. This is specified in the SMF parameter group (enter **/PARMS**). If there is no default SMF record ID and this operand is not specified, the reporting thread fails with an invalid record ID message.

INTERVAL=*hours-between-smf-records*

Specifies the number of hours between the generations of SMF records. This operand is only allowed if NCL CPU-time reporting is inactive at the time the command is entered.

Limits: 0 to 24 (note, 0 is equal to 24). Only numbers which are whole factors of 24 (that is, 1, 2, 3, 4, 6, 12, and 24) are valid.

Default: 24

FROM=*start-time-for-first-report*

Specifies the base time for the calculation of intervals. The default if this operand is not entered is 00.00.00 (midnight). The time that the first SMF record is generated is calculated by adding the hours-between-smf-records value to the start-time-for-first-report value.

For example, if the hours-between-smf-records value is 4 hours, and the start-time-for-first-report value is 01.10.00 (1.10 am), and the command is entered at 08.00.00 (8.00 am), then the next report will be at 09.10.00 (9.10 am). This is because, using the base time and interval, record generation occurs at 1.10, 5.10, 9.10, 13.10, 17.10, 21.10, and 1.10 again. So, if the command is entered at 8.00 am, the next report is due at 9.10 am.

MINCPU=*minimum-CPU-time-to-report*

Specifies the minimum amount of CPU-time that must have been accumulated since a report was last generated for a user, before another report is generated for that user. It is effective immediately after the command is issued, and is applied on the next report.

Limits: A value in CPU seconds, with up to 2 decimal places (for example, 1.25 seconds).

Default: 0

SET

Modifies the NCL CPU-time reporting parameters. It can be specified whether CPU-time reporting is active or inactive; however, the FROM and INTERVAL operands are accepted only when CPU-time reporting is inactive.

STOP

Stops user NCL CPU-time reporting. This operand controls whether SMF reports on the statistics are generated. The actual accumulation of user CPU-time statistics occurs independently of the reporting process, thus CPU-time statistics are collected irrespective of whether reporting is active or inactive.

This operand is accepted only if reporting is currently active. No other operands are valid in conjunction with STOP.

Note: NCL CPU-time accounting occurs independently of the report generation process. A CPU-time report for a user contains a field which shows the time at which the last report for that user was generated.

Display the Status of NCL CPU-time Accounting

The SHOW USERACCT command displays the current status of user NCL CPU-time accounting.

The information displayed shows the following:

- Whether NCL CPU-time accounting is active
- The SMF record ID
- The start time for reporting
- The SMF reporting interval
- The CPU-time threshold for reporting

The start time is in the form *hh.mm.ss*, the interval is in hours, and the threshold is in hundredths of a second.

Chapter 5: Administering Signed-on Users

This section contains the following topics:

Show Active Users

At any time, only a selection of the defined users are actively signed-on to, or disconnected from, a given system. You can display these user IDs using *one* of the following methods:

- SHOW USERS command in the OCS window
- Active User List Facility

Note: All users must be defined to the security system by using *one* of the following:

- The User ID Access Maintenance Subsystem (UAMS)
- An external security package

For more information about security, see the *Security Guide*.

SHOW USERS Command

To show active users

1. From OCS, enter **SHOW USERS** at the prompt.

   Note: For more information about this command, press F1 (Help).

Active User List Facility

The Active User List facility assists help desk staff or the systems administrator to perform the following functions for one or more domains connected by INMC links:

- Monitor user activity
- Cancel a user
- Disconnect a user

The System Support : User ID List lets you identify which domain (in the generic resource) each user is currently attached to. You can then apply any of the available actions to a particular user attached to a particular domain.

On the User ID List, entries are displayed within their domains, with delimiter lines dividing the domains. The local domain (if selected) is always shown first, with others following in domain name (link name) order.

Error messages, such as NO MATCHING USER(S) ON THIS DOMAIN, are displayed for any domain where they apply.

Matching user IDs show the user ID and one of the following:

- The terminal name (LU name) if logged on (signed-on user)
- The disconnection data if applicable (disconnected user)

To list active users

1. Enter **/SS** at the prompt.

   The Security and System Services : Primary Menu appears.

2. Type **LU** at the Select Option prompt and complete the following fields:

   User ID

   Specifies the user ID for which you want to search. You can enter the leading characters of a user ID to limit the search. If you enter eight characters, then this value is used as an exact match. If you enter less than eight characters, then this value is treated as a prefix.

   Note: If the last character is an *, then it is ignored; that is, user IDs USER01 and USER01* are equivalent.

   Link or Domain Name

   Identifies the name of the domain from which you want to obtain information. There are four options:

   - Leave blank for the local system.
   - Enter a specific link name.
   - Enter **?** to display a list of link names from which you can select one or more.
   - Enter ***** to denote all active remote domains.

   Note: Link name on this panel means an INMC link that is currently active.

   Press Enter.

   The System Support : User ID List appears. The panel provides actions that enable you to disconnect or cancel a user session.

Chapter 6: Customizing and Using MAI-OC

This section contains the following topics:

MAI-OC

MAI-OC lets you start multiple sessions with VTAM applications using Logical Unit Type-1 (LU1) protocols. It is available from OCS or from an NCL procedure. MAI-OC appears to the application as a line-by-line device, such as an IBM 3767 terminal.

You can use MAI-OC to provide centralized network operation of major systems such as CICS, IMS, or JES. The MAI-OC sessions act as the master consoles of the other application systems.

MAI-OC facilities are available from any Operator Console Services (OCS) window; however, before using MAI-OC, you should consider the use of MAI-OC sessions with certain subsystems, cross-domain MAI-OC sessions, and the use of MAI-OC from multiple systems to the same target applications.

You can operate an MAI-OC session from an NCL procedure using standard internal command environment processing through the &INTCMD facility. The NCL procedure can send data across the MAI-OC sessions that it is maintaining and receive output from those sessions.

MAI-OC sessions with other applications can be created from any processing environment. Most things that can be done from a native terminal can be done using an MAI-OC session.

In this section there are references to the commands that control the operation of MAI-OC. The use and syntax of MAI commands are described in the online help.

Define MAI-OC to Target Applications

When an MAI-OC session is established with a target application, MAI-OC emulates an LU-Type 1 device (for example, an IBM 3767) as the secondary end of the session. The target application sees the MAI-OC connection as a standard session with a physical 3767 terminal.

Certain application subsystems, such as IMS, require that every logical unit with which they are to have a session be defined to them before any session with the LU is allowed.

To establish MAI-OC sessions with systems such as CICS or IMS, include the VTAM LU names for MAI-OC to use when requesting the session in the appropriate system definitions. Other relevant information should be included with the VTAM LU names, such as the ability of the logical unit to act as a master terminal, its authority level.

The definition to a subsystem such as CICS or IMS is the same as for a physical 3767 device. See the appropriate manuals for the precise coding requirements for the system that you are using.

Sample LU Definitions

The following sections provide sample LU definitions for CICS, IMS, JES2, and JES3.

Note: All LU names likely to be used to create a session with these applications should be defined as a separate terminal to that application.

For information about definition requirements and to customize the definitions to your own requirements, see the appropriate subsystem guides.

Define MAI-OC to a CICS System

The following definition enables an MAI-OC session to start with a CICS system using an LU name of NMMAO001:

```
DFHTCT   TYPE=TERMINAL                                                 *
         ACCMETH=VTAM                                                  *
         BRACKET=YES                                                   *
         BUFFER=256                                                    *
         BMSFEAT=(noroute,norouteall)                                  *
         GMMSG=YES                                                     *
         NETNAME=NMMAO001                                              *
         OPERID=id                                                     *
         OPERPRI=code                                                  *
         PGESIZE=(12,80)                                               *
         PGESTAT=PAGE,*RELREQ=(YES,YES)                                *
         RUSIZE=256                                                    *
         TIOAL=256                                                     *
         TRMIDNT=term                                                  *
         TRMSTAT=TRANSCEIVE                                            *
         TRMTYPE=3767
```

Define MAI-OC to an IMS System

The following definition enables an MAI-OC session to start with an IMS system using the LU name of NMMTO and providing master terminal authority:

```
TYPE        UNITYPE=SLUTYPE1
TERMINAL    NAME=NMMTO,COMPT1=(CONSOLE,MFS-SCS1),OUTBUF=256
NAME        (lterm,MASTER)
```

The following definition provides IMS support for a terminal named NMMAO001, which could be used by MAI-OC for general operations and transaction execution:

```
TYPE        UNITYPE=SLUTYPE1
TERMINAL    NAME=NMMAO001,COMPT1=(CONSOLE,MFS-SCS1),OUTBUF=256
NAME        lterm
```

Define MAI-OC to a JES2 System

The following definition lets you use MAI-OC on a JES2 system:

```
LOGON1 APPLID=JES2
&MAXSESS=nn
&NUMLNES=nn
&NUMRJE=nn
&NUMTPBF=nn
&MLBFSIZ=256
&TPBFSIZ=256

LINE1       UNIT=SNA

RMT1        LUTYPE1,BUFSIZE=256,LINE=1,CONSOLE,                            *
            NOCMPCT,NOCOMP,SETUPHDR,                                       *
            SETUPINF,WAITIME=1,                                            *
            NUMPR=0,NUMRD=0,LUNAME=NMMAO001
```

In this example, an MAI-OC session started on LU NMMAO001 is automatically assigned to workstation RMT1. If the LUNAME parameter on the RMT1 statement is omitted, a user creating an MAI-OC session can specify the required workstation name in user data on the MAILOGON command, for example:

```
MAILOGON JES2 DATA=RMT1
```

Define MAI-OC to a JES3 System

The following definition lets you use MAI-OC on a JES3 system:

```
COMMDEFN,APPLID=JES3,LU=nn
CONSOLE,JNAME=RJE01,TYPE=RJP,DEST=NONE
RJPWS,N=RJE01,RD=0,PR=0,G=RJE01,AUTO=N,
COMPACT=NO,C=R,LU=NMMAO001
```

In this example, an MAI-OC session started on LU NMMAO001 is automatically assigned to workstation RJE01. If the LU parameter on the RJPWS statement is omitted, a user creating an MAI-OC session can specify the required workstation name in user data on the MAILOGON command, for example:

```
MAILOGON JES3 DATA=RJE01
```

Cross-Domain MAI-OC Sessions

Unless VTAM is configured to support dynamic cross domain definition and adjacent SSCP lookup, for a user in one domain to request an MAI-OC session with a target application running in another domain, the following conditions must be satisfied:

- The appropriate cross domain resource definitions must be filed in the VTAM definition library
- The target subsystem must have the MAI-OC LUNAME defined to it as a valid terminal if required

Log On to Another Application

To log on to another application using MAI-OC, use the MAILOGON command. Logging on creates an MAI-OC session. Sessions can be created with as many applications as you require and multiple sessions can be created with the same application.

For example, to establish a session with a CICS application with the application ID of CICSA, enter the following command:

```
MAILOGON CICSA
```

When the connection is established, you receive an initial message from the application informing you that you are connected.

Note: If the session is established with JES, you do not receive a message to confirm connection.

Create a Session Identifier

Each session you create is given a unique session identifier. This identifier defaults to the name of the application program with which the session is established, but you can choose any 1- to 8-character name.

For example, to change the session ID of your CICS application from CICSA to CICSPROD, enter the following command:

```
MAILOGON CICSA ID=CICSPROD
```

The session identifier is used in all MAI-OC commands, so you should make it as meaningful as possible.

If you create multiple sessions with one application from the same window, you must allocate a unique identifier to each session. If the first session's identifier defaults to the application name, you must specify a unique identifier for subsequent sessions with the same application.

The uniqueness of a session identifier applies only to the primary environment. You can open another window at the terminal and create more MAI-OC sessions using identifiers already used in the first window.

List Established Sessions

To list all active MAI-OC sessions, enter the SHOW MAI command. This lets you list all of the MAI-OC sessions that are established and the identifiers used for each.

Select an LU Name for an MAI-OC Session

Before you can start an MAI-OC session with a target application, the system must allocate the LU name that is to be used to act as the *terminal* end of the session.

The LU name may be allowed to default or a particular LU name may be specified on the LU= operand of the MAILOGON command.

Choose an LU Name From a Pool

If no specific LU name is provided on the MAILOGON command, MAI-OC generates one consisting of the MAIOPREF prefix (set by the LU1 Terminal Prefix field of the EXTAPPLPOOLS parameter group in Customizer) followed by a number in the range 001 - 999. The number chosen is the first that is not already in use by another MAI-OC session.

This technique lets you establish an MAI-OC session without knowing the identity of the terminal that MAI-OC will simulate. It also implies that when the session is established, the LU name used by MAI-OC is unpredictable.

When an MAILOGON request without a specified LU name fails because MAI has chosen an unknown LU name, MAI tries up to five successive LU names (each beginning with the MAIOPREF prefix) before indicating that no MAI-OC LUs are available. This is because a defined MAI-OC LU may have been varied inactive; therefore, appears to MAI the same as an LU that has not been defined.

Choose a Specific LU Name

If you want an MAI-OC session with a target application in which the MAI-OC LU name must be the name of a specific terminal, the MAILOGON LU= operand lets you specify the LU name that MAI-OC is to use.

This technique requires that you have knowledge of the terminal that is to be used on the session, but it also means that the identity of the terminal is predictable.

This facility is necessary to establish an MAI-OC session that has to have particular attributes, for example:

- An IMS system is generated with its IMS Master Terminal (primary operating console) having the LU name of MTO3767P.
- An operator is to operate an MAI-OC session from the OCS window of a terminal, with the MAI-OC session driving the IMS Master Terminal.
- The Operator requests the MAI-OC session with a MAILOGON command, specifying LU=MTO3767P on the command text. MAI-OC will open a VTAM ACB whose APPL name is MTO3767P and which must have been defined to VTAM as an APPL.

Note: An MAI Installation Exit (MAIEX02), if provided, is driven whenever a session request is processed. This exit may override the LU name or prefix, if required.

Log Off an Application

Most MAI-OC sessions can be terminated by sending a logoff command of the type expected by the application. For example, for TSO this would be LOGOFF, for IMS it would be /RCL.

However, some applications (such as JES) do not have a logoff command, or you may have trouble sending the command. In these cases, you can use the MAIDISC command to force the disconnection (see page 71).

If you exit OCS with MAI-OC sessions intact, your region automatically generates MAIDISC commands for all your MAI-OC sessions. This causes lost terminal conditions at the applications for all your MAI-OC sessions. We do not recommend that you use MAIDISC to end TSO sessions because the logoff leaves a reconnect environment pending for a system-defined period.

Disconnect an MAI-OC Session

If you cannot log off an MAI-OC session normally, you can use the MAIDISC command and specify the session ID that you want to disconnect with. For example, to end the session with the CICSA application that was established above, enter the following command:

```
MAIDISC CICSA
```

Interrupt an MAI-OC Session

You can interrupt an MAI-OC session to achieve different effects depending on the application you are connected to. For example, if you have a TSO session established, an interrupt cancels the current operation. If you have an IMS session, it removes the current message from the queue and requests the next one. See the relevant product guide for more information about the effect of an interrupt.

To interrupt an MAI-OC session, use the MAIINT command. For example, to interrupt the session established with CICSA, enter the following command:

```
MAIINT CICSA
```

The MAIINT command can also be specified to generate an attention interrupt to the application by using the TYPE=ATTN operand, and a cancel interrupt by using the TYPE=CNCL operand.

Send Data to an Application

When you have logged on to an application, you may want to send data to the application.

To send data to an application, use the MAISEND command. This command nominates the session identifier of the session over which you want to send the data, followed by the data you want to send. For example, to send data to the CICSA session, enter the following command:

```
MAISEND CICSA CEMT I TRAN
```

MAI-OC may append a new line character to the message (to simulate a RETURN key) and the data is sent.

Note: An MAI-OC session looks like a session with a hard-copy terminal. It does not function on a full screen basis. For example, an attempt to invoke ISPF on a TSO session is rejected.

Receive Data from an Application

Data received from an application is issued as line messages to the environment that last issued an MAI-OC command against that session. For example, if you issued an MAILOGON command from OCS, then the initial application messages resulting from the session establishment are received by the OCS environment. If the next MAI-OC command for that session is from a dependent environment, for example MAISEND through $CMDENT, then further messages are returned to the dependent environment.

Application data is displayed unchanged, with the possible addition of some information before the text. This information is the session identifier of the session from which the data was received. For example:

```
(CICS) H2002I TERMINAL CONNECTED
```

The presence of the prefix information and its format is controlled by options specified on the MAILOGON command.

The messages may be in response to a command or NCL system variable that was issued, or they might be unsolicited information, depending on the way the application functions. However, all messages are flagged as unsolicited.

Note: The PROFILE UNSOL=NO command does not prevent the receipt of messages generated by an MAI-OC session.

Issue Commands to an Application

When you issue commands to an application using an MAI-OC session, it works in the same way as issuing commands using OCS. However, there are some special considerations for sessions with IMS applications.

MAI-OC lets you simulate a logical keyboard locked condition, as well as abbreviate commands, use function keys, and use NCL procedures to simplify control procedures for the application.

Any commands that can be issued from an OCS window can also be issued from an NCL procedure. Even NCL procedures operating in full-screen mode (for example, invoked through an FSPROC command) can make use of MAI-OC sessions.

Issue Commands While Waiting for Application Response

Because MAI-OC simulates a real terminal, it is possible to get a logical keyboard locked condition in which MAI-OC is, for instance, waiting for a response from the application. At this time, the MAISEND command cannot be used to send data to the application and if entered is rejected with an appropriate error message. Normally the command can be retried later. Of course, any other region commands can be entered while you are waiting.

Abbreviate Commands

NCL procedures and terminal function keys can be used to simplify MAI-OC command requirements, and many MAI-OC commands can be shortened or automated using the EQUATE command.

It is possible that EQUATE commands and NCL procedures have already been set up that you can use. Check with your systems administrator.

Issue Multi-Segment Commands to an IMS Application

When sending commands to an IMS application over an MAI-OC session, IMS requires that some input messages be multi-segment. Specifically, a /BRO command must be in two segments. Consider, for example, the following command:

```
/BRO NODE NMMAV003 COFFEE TIME
```

To make this command form two segments, IMS requires a new line character after the node name and before the message.

The MAILOGON command lets you specify a character to represent a new line character in data sent through an MAISEND command.

For example, you can create an MAI-OC session using the following command:

```
MAILOGON IMS NL=+
```

This command defines the new line character as the plus sign (+). You can now send the above broadcast command by entering the following command:

```
MAISEND IMS /BRO NODE NMMAV003+COFFEE TIME
```

The plus sign (+) is replaced by the necessary new line character.

Use MAI-OC with Multiple Regions

When more than one region in a network has MAI-OC sessions with the same set of target applications, you should assign a unique set of MAI-OC LU names to each region. This associates the names used by each region with the region in which that region is executing, and avoids VTAM definition conflicts when attempting to start cross-region sessions.

The other advantage of assigning each region its own set of MAI-OC LU names is that it allows added control over which regions can establish MAI-OC sessions with which subsystems.

Set EQUATE Values for MAI-OC Commands

The MAI-OC feature performs the functions of session connection and disconnection and sends messages on sessions in response to MAI-OC commands issued by users.

MAI-OC commands can be made easier to use if the standard SOLVE EQUATE command is utilized. For certain applications, use of EQUATE values enables the operation of MAI-OC sessions to be identical to operation of the same session from a native terminal attached directly to the application.

For example, if you have an MAI-OC session with an IMS system and want to display the IMS transaction queue, you would use the following full MAI command:

```
MAISEND IMS /DIS Q TRAN
```

MAISEND

Specifies the command that requests MAI-OC to transfer data across a session.

IMS

Specifies the session identifier of a session with the IMS application to which the message is to be sent.

The remainder of the data, starting with the slash (/), is the message to be sent to the application.

By setting up an EQUATE value as:

```
EQUATE / MAISEND IMS /
```

you can enter the IMS message text in its native form and allow the system to expand the slash (/) to the full MAISEND command format.

An alternative may be to EQUATE the target application name to the MAISEND command.

Another example, for use on a JES session, could be:

```
EQUATE $ MAISEND JES2 $
```

You can use it to enter JES2 commands as if in native mode, for example:

```
$DA
```

Note: The use of EQUATEs varies according to the requirements of different installations, and you should consider how EQUATEs can be used to make MAI-OC operation simple in your installation.

Use an MAI Installation Exit

An MAI installation exit (MAIEX02) is provided with your product. This provides security checking and validates and changes the characteristics of an MAI-OC session.

The exit is driven whenever a session request is to be processed if the MAIEX02 SYSPARM has an exit name specified. This exit can override the LU name or prefix, if required.

Session Protocols

An MAI-OC session functions as a true SNA LU-type 1, and adheres to the protocols described in the IBM publication *3767 Component Description*.

When MAI-OC has a session with JES2 or JES3, it appears as a 3776-type RJE device. This is still an LU-type 1, and the protocols used are a subset of those described above.

As MAI-OC is simulating a real terminal, it is possible to get a logical *keyboard locked* condition in which MAI-OC is, for instance, waiting for a response from the application. At this time, the MAISEND command cannot be used to send data to the application, and if entered will be rejected with an appropriate error message.

The SHOW MAI command can be used to determine the session states of MAI sessions. Information given includes whether the *keyboard* is locked or unlocked, the SNA bracket state, and the general session state. Abbreviations used for states in the display are generally those used in the *3767 Component Description*:

INB

In Bracket

BETB

Between Brackets

BBP

Begin Bracket Pending

SEND

Send State (can send to application)

RCV

Receive State (cannot send)

DRWT

Waiting for a definite response (cannot send)

STBY

Standby State (can send).
Indeterminate state (state change in progress or not in session).

The CON field in the display may contain the following:

YES

Session established and available.

NO

Session not yet established.

LCK

Session established but keyboard locked, because session state is not such that data may be sent.

SCS Character Support

SCS control characters are used by some systems for print layout instructions. They tell an output device (usually a printer) how to respond to tab, spacing, line break and other formatting control sequences.

Not all SCS control characters can be fully simulated at a terminal; however, none cause a session to be rejected, and wherever possible MAI-OC translates the SCS character to the best equivalent function that OCS mode can provide.

This section details the actions taken by MAI-OC on receiving data streams containing the following SCS characters:

New Line X'15'

Data following the character is displayed on a new line of the operator window.

Form Feed X'0C'

As New Line.

Line Feed X'25'

Stripped from the data stream.

Vertical Tab X'0B'

As New Line.

Record Separator X'1E'

As New Line.

Carriage Return X'0D'

As New Line.

Vertical Channel Select X'04nn'

As New Line.

Horizontal Tab X'05'

Replaced by a blank.

Backspace X'16'

Logically deletes previous character in the line.

Inhibit Print X'24'

Stops data sent to the application being echoed to the screen or the activity log. Data is replaced by asterisks.

Enable Print X'14'

Resumes echoing after a previous Inhibit Print.

Set Horizontal Format X'2BC1'

Stripped from the data stream (together with all associated counts and so on).

Set Vertical Format X'2BC2'

As Set Horizontal Format.

SCS Characters Sent by MAI-OC

The only SCS character sent by MAI-OC to an application is the New Line (X’15’) character. It is appended to each message sent and you can be embedded in data. For more information, see the NL= operand of the MAILOGON command in the online help.

Strike-over Masks

A common technique used on hard copy terminals to hide entered data such as passwords is the use of a strike-over mask, where two or more lines of characters are printed one over the other, and the print head left underneath these characters. The next line of data typed is then unreadable.

MAI-OC keeps track of where the print head would be on a real hard copy terminal and prevents the echoing to the screen or activity log of all or some of the next line of data sent to the application. Any characters sent that would be underneath other characters are replaced by an asterisk. For example, suppose MAI-OC received the following string of characters from the application:

```
XXXXXXXX<LLLLLLLL<00000000<
```

<

Specifies an SCS carriage return (X‘0D’).

If the MAI-OC user were then to send the characters MYPASSWORD to the application, those characters would be echoed to the screen and log as ********RD, because the first eight characters would be obscured on a real terminal. Multiple backspace characters instead of carriage return could be used in the mask.

An alternative to the use of strike-over masks is the use of the Inhibit and Enable Print SCS control characters.

JES MAI-OC Sessions

JES regards an MAI-OC session as a session with an RJE workstation. This means that data sent to JES is regarded as input from a remote console, so any authorized JES command may be sent and the results returned to the window.

However, JES does limit the scope of commands that can be entered from a remote console. Generally, with the provision of the appropriate operands on commands and the correct authorization in JES, commands can be entered to perform any JES display-type function. However, commands can only change the status of jobs, and so on, owned by the workstation. Of course, the OPSYS OCS command can be used to enter commands, if required.

JES commands are available to shorten responses to commands, for example, to remove the leading time stamps). Their use should be considered to make the display as neat as possible.

JES2, and JES3 if so configured, do not send a salutation message to a workstation when it logs on. This means that there is no indication that an MAI-OC session request has completed. Use the SHOW MAI command to determine when the session is established.

JES does not have a logoff command. Use the MAIDISC command to terminate a JES session.

MAI-OC Mode Table and Bind Checks

The following logmode table should be assembled and linked into the appropriate VTAM library (for example, SYS1.VTAMLIB in z/OS). It accurately defines MAI-OC session characteristics and results in the most efficient use of a session. It should then be specified on all MAI-OC VTAM APPL statements using the MODETAB=MAIVMODE operand:

```
MAIVMODE  MODETAB

MAIVMDE   MODEENT LOGMODE=MAIVMDE                                          *
          FMPROF=X’03’                                                     *
          TSPROF=X’03’                                                     *
          PRIPROT=X’B1’                                                    *
          SECPROT=X’90’                                                    *
          COMPROT=X’3080’                                                  *
          RUSIZES=X’8585’                                                  *
          SSNDPAC=X’00’                                                    *
          SRCVPAC=X’01’                                                    *
          PSNDPAC=X’01’                                                    *
          PSERVIC=X’010000008000800000000000’
```

The following table shows the checks MAI-OC performs on BIND parameters at session initiation. The bits shown are checked by MAI; bits not shown are not checked. Invalid BIND parameters are rejected by MAI.

Byte	Bit	Setting	Meaning
2	all	X’03’	FMPROF
3	all	X’03’	TSPROF
4			PRIPROT
	2-3	B’00’	Invalid
		B’01’	Exception response
		B’10’	Definite response
		B’11’	Exception or definite response
	6	B’0’	Compression not used
	7	B’1’	End bracket may be sent
5			SECPROT
	2-3	B’00’	Invalid
		B’01’	Exception response
		B’10’	Definite response
		B’11’	Exception or definite response
	7	B’0’	End bracket not sent
		B’1’	End bracket may be sent
6			COMPROT1
		2B’1’	Brackets are used
	3	B’1’	Bracket termination rule 1
	4	B’0’	Alternate code not used
7			COMPROT 2
	0-1	B’00’	Invalid
		B’10’	Flip-flop mode
		B’01’	Contention mode
		B’11’	Invalid
	2	B’0’	Contention loser recovers
	3	B’0’	Primary is contention loser

MAI-OC Logmode Entry Selection

MAI chooses the logmode entry for an MAI-OC session by searching (by name) the logmode table specified by the MODETAB operand in the APPL definition for the MAI-OC LU selected. The logmode table specified by this operand must be assembled and linked into a load library accessible to CA SOLVE:Access.

The MAIVMODE table supplied in the CC2DSAMP distribution library contains a sample logmode entry used by MAI for MAI-OC sessions. It is recommended that this entry be copied into the logmode table specified. Alternatively, the MAI-OC APPL definition may specify the MAIVMODE table. This is the case in the sample MAI-OC APPL definitions.

MAI-OC Operational Scenario

This section contains examples of VTAM and MAI-OC definitions that are necessary in an installation with the following configuration and requirements:

- Two regions, one called NMP running on the production machine, the other called NMT running on the testing machine
- TSO running on both machines, one called TSOP, the other TSOT
- IMS running on both machines, one called IMSP, the other IMST
- A Network Operator uses a terminal logged on to NMP to:
  - Control VTAM in both machines
  - Operate IMS Master Terminals to both IMS systems using MAI-OC from his terminal
- Authorized personnel can log on to either region and create MAI-OC sessions with the TSO or IMS of their choice. A maximum of three MAI-OC sessions from each region are allowed.

Production Machine Definitions

This section contains examples of the following production machine definitions:

- VTAM
- MAI-OC
- IMSP

VTAM Definitions

```
MAOP001 APPL    MODETAB=MAIVMODE,EAS=1
MAOP002 APPL    MODETAB=MAIVMODE,EAS=1
MAOP003 APPL    MODETAB=MAIVMODE,EAS=1
MAOMTOP APPL    MODETAB=MAIVMODE,EAS=1
MAOMTOT APPL    MODETAB=MAIVMODE,EAS=1

MAOT001 CDRSC CDRM=TCDRM
MAOT002 CDRSC CDRM=TCDRM
MAOT003 CDRSC CDRM=TCDRM
```

MAI-OC Definitions

The value of the LU1 Terminal Prefix field in the EXTAPPLPOOLS parameter group is NMMAV.

IMSP Definitions

```
TYPE          UNITYPE=SLUTYPE1
TERMINAL      NAME=MAOMTOP,COMPT1=(CONSOLE,MFS-SCS1),
              OUTBUF=256
NAME          (MAOMTOP,MASTER)
TERMINAL      NAME=MAOP001,COMPT1=(CONSOLE,MFS-SCS1),
              OUTBUF=256
NAME          MAOP001
TERMINAL      NAME=MAOP002,COMPT1=(CONSOLE,MFS-SCS1),
              OUTBUF=256
NAME          MAOP002
TERMINAL      NAME=MAOP003,COMPT1=(CONSOLE,MFS-SCS1),
              OUTBUF=256
NAME          MAOP003R
TERMINAL      NAME=MAOT001,COMPT1=(CONSOLE,MFS-SCS1),
              OUTBUF=256
NAME          MAOT001
TERMINAL      NAME=MAOT002,COMPT1=(CONSOLE,MFS-SCS1),
              OUTBUF=256
NAME          MAOT002
```

TERMINAL	NAME=MAOT003,COMPT1=(CONSOLE,MFS-SCS1),
	OUTBUF=256

NAME	MAOT003

Network Operator Action

The following creates an MAI-OC session to IMSP as Master Terminal:

```
MAILOGON IMSP LU=MAOMTOP NL=+ WAIT=PERM
```

The following creates a cross-domain MAI-OC session to IMST as Master Terminal:

```
MAILOGON IMST LU=MAOMTOT NL=+ WAIT=PERM
```

Other User

The following creates a session with IMSP using LU MAOP001:

```
MAILOGON IMSP
```

The following creates a cross-domain session with IMST using LU MAOP002:

```
MAILOGON IMST
```

The following creates a cross-domain session with TSOT using LU MAOP003:

```
MAILOGON TSOT
```

Testing Machine Definitions

This section contains examples of the following testing machine definitions:

- VTAM
- MAI-OC
- IMST

VTAM Definitions

```
MAOT001 APPL    MODETAB=MAIVMODE,EAS=1
MAOT002 APPL    MODETAB=MAIVMODE,EAS=1
MAOT003 APPL    MODETAB=MAIVMODE,EAS=1

MAOP001 CDRSC CDRM=PCDRM
MAOP002 CDRSC CDRM=PCDRM
MAOP003 CDRSC CDRM=PCDRM
MAOMTOT CDRSC CDRM=PCDRM
```

MAI-OC Definitions

The value of the LU1 Terminal Prefix field in the EXTAPPLPOOLS parameter group is MAOT.

IMST Definitions

```
TYPE        UNITYPE=SLUTYPE1
TERMINAL    NAME=MAOMTOT,COMPT1=(CONSOLE,MFS-SCS1), OUTBUF=256
NAME        (MAOMTOT,MASTER)
TERMINAL    NAME=MAOP001,COMPT1=(CONSOLE,MFS-SCS1), OUTBUF=256
NAME        MAOP001
TERMINAL    NAME=MAOP002,COMPT1=(CONSOLE,MFS-SCS1), OUTBUF=256
NAME        MAOP002
TERMINAL    NAME=MAOP003,COMPT1=(CONSOLE,MFS-SCS1), OUTBUF=256
NAME        MAOP003
TERMINAL    NAME=MAOT001,COMPT1=(CONSOLE,MFS-SCS1), OUTBUF=256
NAME        MAOT001R
TERMINAL    NAME=MAOT002,COMPT1=(CONSOLE,MFS-SCS1), OUTBUF=256
NAME        MAOT002
TERMINAL    NAME=MAOT003,COMPT1=(CONSOLE,MFS-SCS1), OUTBUF=256
```

NAME	MAOT003

User Action

The following creates a session with IMST using LU MAOT001:

```
MAILOGON IMST
```

The following creates a cross-domain session with IMSP using LUMAOT002:

```
MAILOGON IMSP
```

The following creates a session with TSOT using LU MAOT003:

```
MAILOGON TSOT
```

Chapter 7: Using Linked Product Regions

This section contains the following topics:

Communicate Between Your Product Regions

Your system services provide the following methods to communicate between regions:

- INMC—Inter-Network Management Connection
- APPC—Advanced Program-to-Program Communication

If two or more regions are connected, be aware of which regions are connected and ensure that the links are active when required.

Display Links

You can display links from a command entry panel.

To display links

1. Enter the following command at the prompt:

   ```
   SHOW LINKS TYPE=type
   ```

   type

   Specifies the type of link, that is, INMC or APPC.

Note: You can also display INMC links by entering the shortcut **/INMC**. This method displays more information about the links.

Link Regions

The INMC feature lets you establish and operate links between regions.

The DOMAIN command is used to define a remote region. The command includes the following:

- The name of the region definition. This is the same as the name specified by the PRI JCL parameter.
- The access methods that can be used to contact remote regions
- Access method specific details:
  - For VTAM, the region name is used as the VTAM ACB APPL name.
  - For TCP/IP, a region definition that includes the IP name or the address and port number.

The LINK START command is then used to establish the INMC to the remote region identified by the DOMAIN command.

Inter-Network Management Connection

INMC provides a general-purpose data transport mechanism that allows a region to communicate with one or more regions across a logical link known as an INMC link.

Simple INMC Links

INMC provides basic functionality, which is suitable for networks where there is only one physical network path between regions. Simple INMC provides the following functionality:

- It allows a region to communicate with one or more other regions across a logical link.
- It allows a maximum of two sessions between any two regions. Each region uses one session as the transmit part of the link and the other as the receive part.
- You can set up simple link definitions, which are stored in the VFS data set, or you can pass all necessary parameters on link start.

Access Methods

Regardless of the link type, INMC links support VTAM, TCP/IP, XNF, or EPS access methods.

Note: The VTAM application definition that you use for general communications (that is, the APPL identified by the PRI JCL parameter) should include the PARSESS=YES operand. This parameter is required to support the parallel sessions that make up any INMC links that use VTAM.

Security

INMC provides a security exit, allowing verification of the identity of the remote region.

Note: For more information, see the *Security Guide*.

Traffic Flow on Simple Links

Data traffic flows between two domains across the twin-session link between the two regions.

Traffic is directed according to preference, across the session in which the transmitting region is the primary end. Therefore, the link can be regarded as having a transmit part and a receive part for each region. The two sessions operate independently.

If there is only one active session, then the traffic flows across the session in both directions.

Control Links

Links are established by using the LINK START command.

Links are stopped by using the LINK STOP command, and can be reset by using the LINK RESET command.

INMC Links

There are two types of INMC links:

- Static INMC links
- Dynamic INMC links

Static INMC Links

Static INMC links are most useful between regions that regularly communicate with each other and that always use the same region names.

The link between two regions cannot be established unless each region has been defined to the other and is prepared to accept the connection.

Dynamic INMC Links

Dynamic INMC links provide a mechanism for establishing transient INMC connections between regions to satisfy temporary connection requirements, without specifying permanent definitions for particular regions.

Dynamic links are established between pairs of regions. They occur when *both* of the following occur:

- One region actively attempts to establish communications.
- The targeted region is prepared to accept the incoming INMC request on a generic naming basis, without having been specifically instructed to communicate with the other region.

Dynamic link capability is controlled by the DEFLINK command. This command defines the parameters necessary for an INMC link request to be accepted from a remote region without issuing any LINK START commands.

The DEFLINK command specifies the following critical parameters:

- A mask value used to decide whether the INMC link request should be accepted
- A prefix value used to generate the link name of the INMC link that will be created if the request for the connection is accepted

Establish INMC Links

Use the LINK command to define regions to each other.

You can enter the LINK command from OCS or include it in the RMREADY procedure. The LINK command can specify the following to the domain:

- The access method to use (VTAM, XNF, EPS, or TCPIP).
- The link name by which the remote domain is known by this region and by the operators of this region. This can be up to 12 characters long and lets you assign meaningful names to the various regions in your network. It might help if you always assign a link name with the same value as the system ID in the SYSTEMID parameter group for the remote region.
- The name of the region definition that contains the access method details.
- A message prefix to add to all messages received from the remote region.
- The color and highlighting that is to apply to all messages received in OCS from the remote region.
- The retry interval at which your system attempts to contact the remote region following a link outage.

Example: Establish INMC Links

To establish a link with a remote region in Chicago with a region name of CHIC1472 and messages from the remote region displayed in red, blinking, and with a prefix of CH14, issue the following commands:

```
DOMAIN DEFINE CHIC1472 VTAM=YES
```

```
LINK START=CHICAGO DOMAIN=CHIC1472 COLOR=RED HLIGHT=BLINK MSGID=CH14
```

Note: If you are establishing a static link, to start communication between the two regions, you must also enter a LINK command definition to this region from the CHIC1472 region.

Routing Commands

When the link is established you can route commands to the remote region by using the link name.

Example: Routing Commands

To show the users of a remote region called CHICAGO, use the following command:

```
ROUTE CHICAGO SHOW USERS
```

Communication Recovery

If contact is lost on an INMC link, then the action of the regions depends on whether the link is static or dynamic.

Static Links

If contact is lost on a static link, then both the local and the remote regions attempt to re-establish communication automatically.

Dynamic Links

When a dynamic link is established, it operates in the same way as a static link. However, if contact is lost and the dynamic link is broken, then each region acts differently:

- The region that requested the link perceives it as a static link definition and automatically tries to reconnect.
- The region that accepted the link as a dynamic link takes no action to restore the link. It accepts any future request that meets the requirements of a DEFLINK definition.

Control INMC Links

In addition to defining static INMC links, the LINK command is also used to stop and restart links and to delete entire link definitions from storage. You can enter LINK commands from the OCS panel at any time, or you can include them in the READY procedure to have the region attempt link establishment to other regions automatically.

Start an INMC Link

To start an INMC link

1. Issue the following command:

```
LINK START=linkname
```

linkname

Specifies the name by which the remote region is known.

If no previous LINK START command has been issued for this destination, INMC retrieves the link definition stored under the link name from the VFS database and attempts to open an INMC link to the remote region. The number of sessions to be opened and the manner in which they operate is defined in the link definition.

If no link definition exists on the VFS database, INMC accepts the LINK START command and defines a static INMC link (accepting any operands if supplied).

If DOMAIN or APPLID and MSGID operands are included on the LINK START command but a link definition already exists on the VFS database, the extra operands are ignored and the VFS definition used.

Stop an INMC Link

To stop an INMC link

1. Issue the following command:

```
LINK STOP=linkname
```

linkname

Specifies the name by which the remote region is known.

This command terminates all sessions with the remote region identified by the *linkname*, and rejects any attempts by the remote domain to reopen the link. The link remains inactive until a subsequent LINK START command is issued.

Reset an INMC Link

To reset an INMC link

1. Issue the following command:

   ```
   LINK RESET=linkname
   ```

   linkname

   Specifies the name by which the remote region is known.

Note: This command is valid only if the link has already stopped. It removes the current link definition information from storage. A subsequent LINK START command retrieves the definition again from the VFS database, if it exists. A link definition modified through INMC maintenance services can be brought on line only if the current active link definition is first reset.

Display INMC Links

To display the current status of an active link to a remote region

1. Enter **/INMC** at the prompt.

   The INMC : Link Status List panel appears. This lists the sessions linked to the remote region.

2. To view more details about a link's status, enter **S** or **B** beside the link name.

   The INMC : Link Status Display panel appears.

   The information displayed on this panel varies slightly depending on the following:

   - Whether the link is defined and has a description
   - Whether the link has ever been active

Improve INMC Link Performance

To improve the performance of your INMC link, you can make the following changes:

- Increase the size of the INMC internal transmission buffers.
- Increase the RU size specified in the logmode table definition (VTAM only).

Increase Transmission Buffer Size

If you are using a high-speed link to carry INMC traffic, then use the SYSPARMS INMCBFSZ operand to increase the size of the principal INMC transmission buffers. The default size is 4 KB; the maximum size is 15 KB.

Example

To increase the size of the transmission buffer to 8 KB, enter the following command:

```
SYSPARMS INMCBFSZ=8
```

Increase RU Size in Mode Table Definitions

Note: This method applies only to links that use the VTAM access method.

The RU size specified on the BIND parameters when INMC sessions are established should match or exceed the size specified by the SYSPARMS INMCBFSZ operand.

If you use SYSPARMS INMCBFSZ to increase the INMC internal buffer size, you must also specify an RU size of at least the size of the INMC buffers for the sessions that are established. If you do not do this, you will not gain any benefit from the increase. If you specify RU sizes greater than the INMC internal buffer size, then the excess is not used.

The RU size is specified by the RUSIZES parameter of the logmode table entry.

The minimum RU size permitted for INMC sessions is 256 bytes. The recommended RU size is 1 KB. If no RU sizes are specified or if a value less than the minimum size is set, INMC uses a default value of 256 bytes.

Use Security with INMC Links

INMC provides a security exit that allows you to implement security to determine whether a link between two regions should be established.

An assembler exit point is provided by INMC to pass control to an installation-coded routine. This routine communicates with an equivalent routine at the remote region and exchanges any identification information that might be required.

For information about the parameter lists and coding requirements for the INMC security exit, see the *Security Guide*.

Diagnose Problems with INMC Connections

If problems occur when establishing INMC sessions between regions, then a tracing facility is provided to help you determine problems. This is provided by setting the SYSPARMS LNKTRACE and SESSMSG operands.

Troubleshooting for INMC Links

For any given pair of regions with defined links, establishing a connection should be automatic once there is a path to each region through the network.

If a link cannot be activated, check the following:

- Has each region been defined to the other? If not, see your systems administrator.
- Does a SHOW LINK display in either region have a link status of PEND-ACT? If so, check that both regions are active, and check the status of the appropriate VTAM cross-region resource managers and resources.
- Does a SHOW LINK display in one region indicate a link status of ACTIVE, but show as STOPPED or PEND-ACT in the other region? If so, this is a system error and you should see your systems administrator. The link definitions for both regions should be stopped, reset, and redefined.
- Does a SHOW LINK display give the status in one region as PEND-ACT, and as STOPPED in the other region? If so, issue a LINK START=*linkname* for the INACTIVE definition.
- If both regions indicate a status of PEND-ACT for the link while there appears to be a network path open between the regions, see your systems administrator.
- Does a SHOW LINK command in either region have a link status of FAILED? If so, check backwards through the activity log for error messages recorded at the time of the failure. Refer these to your systems administrator. The link can be restarted using a LINK START command.

Centralized Control of Connected Regions

The Remote Operator Facility (ROF) provides centralized control, at the user level, of regions connected using INMC.

Use ROF for Centralized Control of Remote Regions

ROF lets you monitor and control remote regions from the local region through OCS. ROF uses INMC as its transport facility. ROF services are available only between two regions when there is an active INMC link between them.

Define User IDs for ROF

When defining a user ID so that it can be used to operate ROF sessions, the following questions should be considered:

- What authority level does the user need to issue the commands required to control remote regions?
- What NPF message and resource partitioning is to apply to the user?
- What types of messages are required for them to receive, and how are these messages to be delivered?
- Is the user ID defined on all the remote regions to which it can log on?
- Do the regions have a separate or shared UAMS data set?

Display ROF Users

To identify which users are operating ROF sessions, either to or from remote regions, use the SHOW USERS command.

Include Simplified Command Definitions in INIT and READY

The EQUATE command is used to simplify long and complicated commands that ROF operators have to issue frequently. You can use the EQUATES parameter group (enter **/PARMS**) to make EQUATE commands available as system-wide defaults.

Use ROF with NCL Processes

Any NCL procedure started by a ROF operator can be used to issue commands to remote regions. The results of these commands are returned to the OCS window of the ROF operator who started the procedure.

&INTCMD can also be used from in an NCL procedure to issue commands to remote regions. The results from these commands are returned to the response queue of the issuing procedure and can be read using &INTREAD.

Use ROF from Background Environments

Background environments can create ROF sessions with remote regions. When a ROF session is started, a standard ROF signon occurs using the background environment’s user ID. If this user ID is not defined to the remote region then the signon fails.

Background environments can route commands to a remote region over a ROF session and have the results returned to it for examination.

Use ROF from the System Console

System consoles can establish ROF sessions with remote regions and issue commands to that domain. If your console user ID has a UAMS definition, then the user ID must be defined on the remote region if a ROF session is to be established.

If you are using the default console user ID (.DFLTOP), then ROF sessions are not supported.

Sign on to a Remote Region

To sign on to a remote region, use the SIGNON command.

Example

To sign on to the remote region PROD01, enter the following command:

```
SIGNON PROD01
```

The remote region checks to see if you are authorized and then notifies you of the connection.

When the connection is established, you receive any unsolicited messages from that region.

Sign on to a Remote Region over an Inactive Link

If there is no active INMC link to the remote region you want to sign on to, you can issue a SIGNON command to wait in a queue for the INMC link to become active.

Example

To sign on to PROD01 but the INMC link to it is not active, enter the following command:

```
SIGNON PROD01
```

This places your signon in a queue and waits until the INMC link to the remote region becomes active. Once the link becomes active, the ROF session is established and you are notified of this by the system.

All ROF sessions and queued signons are canceled when you exit OCS.

Restart a ROF Session

If an INMC link to a remote region fails while you are connected, your ROF sessions to that region are terminated and queued for reestablishment when the link becomes active again. You are notified when the ROF session is reestablished.

If you exit OCS before the link becomes active, the queued SIGNON is canceled.

Issue Commands on a Remote Region

To issue a command on the remote region, use the ROUTE command.

Example

To send a SHOW USERS command to PROD01, enter the following command:

```
ROUTE PROD01 SHOW USERS
```

The SHOW USERS command executes in PROD01 as if it had been entered from a local terminal, and the result of the command is returned to your OCS window.

Note: If you route a command to a remote region that you are not signed on to, a signon to that remote region is automatically initiated.

Use Command Separators

The ROUTE command allows support for a single embedded colon (:) character as a command separator. When the ROUTE command processor encounters a single colon in the command string, it substitutes a semicolon. If the processor encounters two colons, it eliminates the first and sends the second to the target region as part of the transmitted command.

Simplify Remote Command Execution

Sending commands to remote regions for execution can be simplified by using EQUATE command strings. For more information about the EQUATE command, see the online help.

Issue Commands from a Remote Region

You can issue a command on another remote region through your current ROF session. This lets you establish ROF sessions to remote regions through any number of intermediate regions.

Example

To send a SHOW USERS command to PROD02 through PROD01, enter the following command:

```
ROUTE PROD01 ROUTE PROD02 SHOW USERS
```

This command sends the SHOW USERS command to PROD02 through an intermediate ROF session with PROD01. This provides an alternative to a direct ROF session, and can be useful if direct contact cannot be established with the target region.

Receive ROF Messages

Whenever you receive messages from remote regions, whether as unsolicited information or in response to a command, each message is color coded and prefixed with a message ID.

Use Color Coded Messages

The messages from remote regions can have different colors depending on which remote region they originate from. This helps you to identify the source of a message if you are connected to multiple remote regions.

The SIGNON command lets you override the default color/highlight attributes which were specified when the INMC link to the remote region was established.

Example

To specify the color red for any messages received from PROD01, enter the following command:

```
SIGNON PROD01 COLOR=RED
```

Use the Message ID

By default, ROF messages returned from a remote region across an ROF session are prefixed with a 1- to 4-character identifier of the remote region specified when the INMC link was established.

The SIGNON command lets you override the ROF message ID prefix default. This prefix can be any 1- to 8-character string.

Example

To specify the message ID for any messages received from PROD01 as TEST, enter the following command when you sign on:

```
SIGNON PROD01 ID=TEST
```

Your prefix, and not the default PROD01, appears as the first word of each ROF message from the remote region.

Such prefixes are private to you and can be varied for each of your OCS windows, or be different for each SIGNON command issued by NCL processes executing in your NCL processing region.

Sign Off a Remote Region

To sign off a remote region, use the SIGNOFF command.

Example

To sign off PROD01, enter the following command:

```
SIGNOFF PROD01
```

This command is used for explicit signoff; however, if you exit OCS, all ROF sessions terminate automatically.

Manage APPC Links

APPC lets you establish links between regions and between a region and other applications. If you have NCL applications that use APPC, you must create APPC links.

Start APPC Links

To start an APPC link, use the LINK command and specify the type of link, with the link name of the region or application you are connecting with.

Example

To establish an APPC link with NMA from SOLVE01, enter the following command:

```
LINK TYPE=APPC START=NMA
```

Sometimes you may have to supply a password before an APPC session is established. This is set up in the APPC link definition by your system administrator.

APPC over an INMC Link

You can start an APPC link that uses INMC as its access method (transport provider).

To start this kind of APPC link

1. Enter the following command:

   ```
   LINK TYPE=APPC START=NMA AM=INMC
   ```

   The link is not established until the INMC link to NMA is active.

Note: Because an INMC link can use TCP/IP as its transport provider, starting an APPC link over an INMC link means that you can establish an APPC link where there is no physical VTAM network connecting the two regions.

Display APPC Link Status

The SHOW LINK command is used to display APPC link status.

Example

To find out the status of all APPC links in your region, enter the following command:

```
SHOW LINK TYPE=APPC
```

The following information about APPC links is displayed:

- Link Name
- Remote LU Name
- Link status
- Link type (parallel or single session)
- LU6.2 options supported (for example, mapping)
- Link and conversation level security supported
- The session limit for the link

Stop APPC Links

An APPC link can be stopped using the LINK STOP command.

Example

To stop the APPC link between SOLVE01 and NMA, enter the following command:

```
LINK TYPE=APPC STOP=NMA
```

Chapter 8: Broadcasts

This section contains the following topics:

Broadcast Services

Broadcast Services let you utilize the various broadcasting capabilities of your system services. Broadcast messages can be sent to terminals, users, and applications, and can be stored on a file.

Broadcast Services let you send the following types of broadcast:

- A general broadcast of one to four lines
- A primary menu broadcast of one line
- A broadcast of up to four lines to all, or specific, EASINET terminals
- A broadcast of up to four lines to all, or specific, system services attached terminals (including EASINET terminals)
- A broadcast of up to four hundred lines to all, or specific, users
- A broadcast of up to four hundred lines to a selected list of users.
- A broadcast of up to four lines to MAI users of an application (for example CICS or IMS).
- A broadcast of up to four lines to users of an NCL application.
- A broadcast to a specific user ID according to their preferred method of notification (as indicated in their UAMS security profile).

Note: Broadcast capabilities are also provided by the $BSCALL NCL interface. For more information, see the *Network Control Language Reference Guide*.

Types of Broadcasts

The broadcasts listed above can be grouped into the following types:

- General
- Primary Menu
- User

These broadcasts are either static or dynamic.

List Broadcasts

To display a list of all active broadcasts in the system

1. Enter **/BCAST** at the prompt.

   The Broadcast Services : Primary Menu appears.

2. Enter **L** at the prompt.

   The List Broadcasts panel appears.

   Note: The broadcasts in the selection list are either permanent or still to be viewed. A permanent broadcast is displayed until deleted by a user. Other broadcasts are displayed until they have been viewed by all target recipients.

View Active Broadcasts

To view the contents of an outstanding broadcast

1. Enter **S** (Browse) next to the required broadcast in the selection list.

   The details appear.

Delete Active Broadcasts

To delete an active broadcast

1. Enter **D** (Delete) next to the required broadcast in the selection list.

   The broadcast is deleted.

   Note: If you want to delete a broadcast before all target recipients have received the broadcast, enter **FD** (Force Delete) next to the required broadcast in the selection list.

Broadcast to Generic Resources

This section describes how to use the Broadcast Services facilities to enable broadcasts to regions belonging to a VTAM generic resource.

These facilities are available from the Broadcast Services : Primary Menu (**/BCAST**).

Enable Broadcasts to Generic Resources

To enable user and terminal broadcasts to regions belonging to a VTAM generic resource group, there must be a broadcast system group corresponding to that generic resource.

To add a broadcast system group corresponding to a generic resource

1. Enter **/BCAST** at the prompt.

   The Broadcast Services : Primary Menu appears.

2. Enter **LS** at the prompt.

   The Broadcast Services : Group List panel appears.

3. Press F4 (Add).

   The Broadcast Services : Group Definition panel appears.

4. Complete the following fields:

   Group Name

   Specifies the name of the new broadcast group.

   Description

   Describes the new broadcast group.

   Include Local System?

   Specifies whether the local system is included in the broadcast group. YES indicates that broadcasts can be issued on this system. NO indicates that broadcasts can be issued on remote systems only.

   Press F4 (Save)

   The group definition is saved. A message is returned, confirming that the new broadcast group has been added.

Add Resources to a Broadcast System Group

To add resources to a Broadcast System Group

1. From the Broadcast Services : Group Definition panel, press F5 (Resources).

   The Broadcast Services : Resource List panel appears.

2. Press F4 (Add).

   The Broadcast Services : Resource Definition panel appears.

3. Complete the following fields:

   Resource Type

   Specifies the type of resource to add. The following types of resource are supported:

   APPCLINK

   A predefined name of an APPC link between two regions.

   Limits: 12 characters

   DOMAIN

   A region (with a name of up to four characters) attached by INMC.

   Limits: 4 characters

   LU

   A network resource name that identifies the required region.

   Limits: 8 characters

   Note: When adding DOMAIN or LU resources, ensure that the necessary DEFLINK commands have been issued to allow the region to connect by using APPC.

   Resource Name

   Specifies the name of the resource to add. This name is used on the Broadcast Services : Send Menu when sending a broadcast to the system group.

   Press F3 (File).

   The changes are saved. The Broadcast Services : Resource List panel appears, with the new resource added.

4. Repeat steps 1 through 3 for each resource that you want to add to the broadcast group.

Send Broadcasts to Generic Resources

When you have defined a broadcast system group for your generic resource, you can use that system group to send broadcasts to regions belonging to the generic resource.

To send broadcasts to generic resources

1. Enter **/BCAST** at the prompt.

   The Broadcast Services : Primary Menu appears.

2. Enter **S** at the prompt.

   The Broadcast Services : Send Menu appears.

3. Complete the following field:

   System Group

   Specifies the system group name for your generic resource.

   Enter the mnemonic of the type of broadcast that you want to send at the prompt.

   The broadcast is sent.

Set General Broadcasts

A general broadcast allows you to notify potential users of the system about critical events. For example, the impending unavailability of a major application such as production CICS or IMS subsystems.

A general broadcast can be up to four lines, and can be sent to all EASINET terminals or all system services terminals, including EASINET terminals. The lines of broadcast appear at the bottom of the terminal display and are available across system restarts.

When you send a general broadcast, the NCL variables &BROLINE1 to 4 are updated, and any panel containing these variables displays the text you have entered the next time those panels are accessed by a user.

To set a general broadcast

1. Enter **/BCAST** at the prompt.

   The Broadcast Services : Primary Menu appears.

2. Enter **G** at the prompt on the Broadcast Services : Primary Menu.

   The Broadcast Services : Edit Broadcast Text panel appears.

   On entry to this panel, the lines in the editor display the text from the last general broadcast that was issued. Details of the user who issued the last broadcast are also displayed.

3. Enter the text that you want to send as a general broadcast in the text lines provided and press F3 (File).

   The broadcast is saved.

   Note: This sets the broadcast so that it can be displayed, and saves the broadcast in the system services VFS file so it is available across system restarts.

Set Primary Menu Broadcasts

The primary menu broadcast allows you to set one line of text to display on the primary menu. This type of broadcast is useful for reminding users of something about the system they have logged on to. For example, you may want to remind users that the system is a test system and that they should not change anything.

To set a primary menu broadcast

1. Enter **/BCAST** at the prompt.

   The Broadcast Services : Primary Menu appears.

2. Enter **P** at the prompt.

   The Broadcast Services : Edit Broadcast Text panel appears. This looks similar to the edit panel for a general broadcast except there is only one text line available for input.

   When you first access this panel, details are given of when the text was last updated, and by which user ID.

3. Enter the text you want displayed on the primary menu in the text line, and press F3 (File).

   The broadcast is saved.

   Note: When you enter text in this line, the NCL variable &ZPMTEXT1, which is contained on the primary menu, is updated. When you press F3, the primary menu is updated and the broadcast is displayed when a user next accesses that menu. F3 also saves the broadcast so that it is displayed across system restarts.

Review Active Broadcasts

To review active dynamic broadcasts that are applicable to your terminal and user ID

1. Enter **/BCAST** at the prompt.

   The Broadcast Services : Primary Menu appears.

2. Enter **R** at the prompt.

   The Broadcast Services : Review Broadcasts panel appears. This panel displays all the broadcasts for your user ID and terminal in the same way you receive a broadcast when you sign on. To view subsequent broadcasts, press F8 (Forward).

   The dashed line above each broadcast on the review panel, includes a message which indicates whether the broadcast is deleted or whether it has been retained by the review function. Retained broadcasts can be deleted from the List Broadcasts panel only.

Send Dynamic Broadcasts

A dynamic broadcast allows you to send a message to a specific user, or to a range of users. When a broadcast is sent, a panel is displayed containing the broadcast lines and details of the broadcast initiation.

To send a dynamic broadcast

1. Enter **/BCAST** at the prompt.

   The Broadcast Services : Primary Menu appears.

2. Enter **S** at the prompt.

   The Broadcast Services : Send Menu appears.

   For information about the fields and options available on the prompt on the Broadcast Services : Send Menu, press F1 (Help).

Enter Broadcast Text

When you have selected the type of broadcast you want to send, and you have set the appropriate fields on the Send Menu, press Enter to invoke the Broadcast Services : Edit Broadcast Text panel. If you select option U, you can specify a maximum of 400 lines of broadcast text. If you select any of the other options, you can specify a maximum of four lines of broadcast text.

Note: If you specified option U and prompting in the Destination Mask field, you are presented with a list of user IDs. You must select the users you want to send the broadcast to and then press Enter to invoke the Broadcast Services : Edit Broadcast Text panel.

When you have entered the broadcast text, press F3 (File) to initiate the broadcast. An acknowledgment message is displayed when processing is complete. This message shows the total number of terminals and users that have received the broadcast at that point in time.

Note: If you send a broadcast to a user who is logged on more than once, the acknowledgment message includes each of their logons in the total number of user IDs that have received the broadcast. However, when the user has viewed the broadcast, the broadcast is discarded for their duplicate logons.

After a broadcast is sent, it is retained until you exit broadcast services. If you want to send another broadcast, the previous broadcast is displayed on the editor if you have not left broadcast services. This allows you to send the same broadcast again, or edit it to send to another user, or group of users.

Receive Broadcasts

When you receive a broadcast that has more lines than the screen has available, you can use the F8 (Forward) and F7 (Backward) keys to scroll through the broadcast.

To acknowledge receipt of the broadcast, you must press F3 (Exit). However, when you press F3 (Exit), you no longer have access to the broadcast. If the broadcast is more than one screen, you can press F3 (Exit) before you have read the whole broadcast.

Access System Group Definition

Broadcast services can be customized to send messages to one or more remote regions. This customization is performed by setting up system group definitions.

Each system group definition contains one or more remote regions to which a broadcast is sent. By specifying a system group definition when sending a broadcast, that broadcast is sent to the remote regions defined in that group definition. The local region can also be included in the system group definition.

The List System Groups option of the Broadcast Services primary menu is used to access the list of defined system groups on your system. From this list you can add, update, and copy system group definitions.

To access a list of the system groups defined

1. Enter **/BCAST.LS** at the prompt.

   The Broadcast Services : Group List panel appears.

   If no system groups have been defined, no entries appear in this list.

   For information about the information displayed and the actions available on the Broadcast Services : Group List panel, press F1 (Help).

Appendix A: Connecting Terminals

This section contains the following topics:

Supported Terminals

The IBM devices listed below are supported by your product region. Any device compatible with those listed is also supported.

IBM Device or Equivalent	Supported Functions
3270 Family LU0, LU2 (up to 255 x 255)	Full function
5550 (24 x 80)	Full function, including DBCS
3767 (or other LU-1 device)	OCS, printing

Information that allows your product to connect to these terminals is contained in their VTAM mode table definition.

Note: Your product region supports full operator functions through the operating system console by use of the MODIFY command. Privilege and authority available through the console can also be controlled. For more information about defining a system console, see the *Security Guide*.

How Your Product Accepts Terminal Connections

When a request for a terminal connection is made to your product, the following process occurs:

1. The logon exit is driven.
2. The logon exit examines the session parameters to determine the terminal type.
3. If the connection is accepted, the session parameters are modified, if necessary, and an OPNDST macro is issued. A BIND request then flows to the terminal. A CLSDST macro is issued if the request is rejected and information about the reason for the failure is sent to the terminal.
4. The terminal receives the BIND and if the parameters contained therein are suitable, logon proceeds and the session is established.

Extended Attributes Support

If you want to specify additional attributes for your terminal connections, you can specify a parameter in the logmode definition that allows your region to send a query to a requesting terminal.

The query obtains the following information from the requesting terminal:

- Extended (seven) color
- Extended highlighting
- Field outlining
- DBCS (double-byte data streams for special language support)
- Screen size

A query is sent when X‘80’ is specified in the second position of the PSERVIC field of the logmode definition, for example, PSERVIC=X‘0280...’.

Specify Screen Sizes

Some terminals and controllers support a special screen size definition in the logmode table definition called *unspecified viewport size*. When unspecified viewport size is specified, the screen size is determined from the parameters received from the terminal in a query.

If unspecified viewport size is not specified, the screen size for the terminal connection is determined by the default or alternate screen size for the terminal type. The default for all terminal types is 24 rows by 80 columns, and the alternate screen size is the largest screen size of which it is capable.

To enable unspecified viewport size, specify X‘03’ in the eleventh position of the PSERVIC field of the logmode table definition.

Notes:

- The session parameters are not altered to specify unspecified viewport size unless the definition also supports a query of extended attributes.
- The screen size for non-SNA terminals is always set using the information obtained by the query, regardless of the setting of the screen size information in the bind.
- An unspecified viewport size BIND fails if sent to a terminal that does not support it.
- The IBM 3174 controller supports an unspecified viewport size bind, as does a 3274 controller at Configuration Support D, Release level 65 and most distributed function devices.

Specify Logmode Table Definitions

The following table specifies the logmode table parameter values that are expected for different terminal types:

Terminal Type	Expected Parameter Values
LU01	FMPROF=X‘02’

LU02	FMPROF=X‘03’
	PSERVIC=X‘01...’

LU03	FMPROF=X‘03’
	PSERVIC=X‘02...’

Examples of suggested mode table entries for most 3270 panels can be found in the MODETABS member in the base installation library CC2DSAMP. In most cases, the default VTAM-supplied tables are adequate.

A sample mode table definition for use with LU-1 terminals that are to be supported by MAI-OC is also supplied in the member.

Note: For more information about coding mode table entries, see IBM's *Communications Server SNA Resource Definition Reference*.

Appendix B: Product Region JCL Parameters

This section contains the following topics:

Product Region JCL Parameter Descriptions

The following list provides detailed descriptions of the JCL parameters that the started task member of your product region can specify. The default value is underlined.

ARMNAME=*name*

Specifies the name used to register a region with the Automatic Restart Manager (ARM). If not specified, the default is SVM_*acbname* where *acbname* is the ACB name set by the PRI= JCL parameter. Use the XOPT= JCL parameter to set ARM options.

DBCS={ <u>NO</u> | YES | IBM | FUJITSU }

Specifies the DBCS option.

DSNQLCL

Specifies the name of the local VSAM data set qualifier. This value is used when allocating data sets.

DSNQLNV

Specifies the name of the local non-VSAM data set qualifier. This value is used when allocating data sets.

DSNQSHR

Specifies the name of the shared VSAM data set qualifier. This value is used when allocating data sets.

DSNQSNV

Specifies the name of the shared non-VSAM data set qualifier. This value is used when allocating data sets.

DYNVVOL=*value*

Specifies a default volume for dynamically allocating files. If this parameter is not specified, no volume is specified in the dynamic allocations and SMS allocates the files according to its rules.

GENRSRC

Specifies the VTAM generic resource name use during initialization of your region. This name is used if you want to implement CA SOLVE:Access as a generic resource in a parallel sysplex.

Note: For additional information about configuring your region to run in a VTAM generic resource environment, see the *Installation Guide* and the *Administration Guide*.

INIFILE=*filename*

Specifies the INI file to use for customization of parameters.

INIRESET

Specifies that all current parameters are to be ignored and the defaults used. You must not use this if an INI file is specified.

INIT={ NMINIT | *name* }

Specifies the name of the NCL procedure, in the COMMANDS procedure library, which is executed as part of system initialization before VTAM ACBs are opened.

Limits: *name* must be 1 to 8 characters.

INIWTO

Specifies that initialization messages are echoed to the system console.

INT=00/*nn*

Specifies which internal commands can be used. The parameter value comprises one hexadecimal byte, where each bit indicates acceptance of a particular command, as shown in the following table. The recommended value is INT=E4, that is, all commands shown in the table.

Bit	Command	Function
X'80'	SHOW STOR	Storage display
X'40	##AT, SH ##AT	AT-trap facility
X'20	##DP	Storage alter
X'10	(reserved)	

Bit	Command	Function
X'08	(reserved)	
X'04	##PMON	Performance monitor
X'02	(reserved)	
X'01	(reserved)	

NCLFM={ * | a }

(z/VM) Specifies the CMS filemode that applies to loading NCL procedures. This must be a valid CMS filemode.

NMDID=*domainid*

Defines the domain ID for this system. If the domain ID is not specified, the first characters (up to a maximum of 4) of the system primary ACB name, specified in the PRI parameter (or the default if not specified), are used as the default value.

The domain ID should be unique across all connected systems. If the value is not unique, it can produce naming conflicts, which restrict inter-system functionality, such as ROF sessions from background environments. Do not confuse this parameter with the system ID in the SYSTEMID parameter group.

Limits: One to four characters

NMSUP=*userprefix*

Specifies the System User Prefix that is used to prefix the user IDs for background system environments. If not specified, the domain ID is used. If the domain ID has not been specified, the first characters (up to a maximum of 4) of the system primary ACB name specified in the PRI parameter are used as the default value.

Limits: One to four characters

NOMODIFY

Specifies that communication between the operating system MODIFY facility and the product, using the system console, is not supported. If this operand is omitted, the use of MODIFY is supported.

NPF={ NO | YES }

Controls the use of NPF. Specify NO to inhibit NPF in the region. This is useful on test regions where security checking is not required but where a production UAMS is used.

NPFFM={ * | a }

(z/VM) Specifies the CMS filemode that applies when loading NPF tables. This must be a valid CMS filemode.

OPT=00/*nn*

Activates CPU accounting support. The parameter value comprises one hexadecimal byte, where each bit indicates the level of accounting, as shown in the following table, which lists the values of the OPT parameter.

Bit	Accounting
X'02'	CPU accounting for all threads
X'01	CPU accounting for NCL threads only

OSINP={ YES | NO }

Controls access to the OSCNTL data set. Access can be either read-only or update. Specifying YES (the default) indicates that the data set is opened for read only access. Specifying NO indicates that it is opened for update. When NO is specified, and an attempt to open the file for update fails, the region retries the open for read only access.

PRI={ NM | *acbname* }

Specifies the name of the primary VTAM ACB for the region to be used for terminal communication.

Limits: One to eight characters

Note: If this parameter is omitted, the default (NM) is also the default for the NMDID and NMSUP parameters, and the system ID in the SYSTEMID parameter group.

PROD={ *product* | (*product1,product2,...productn*) }

Specifies the list of product name key values for the features to include in the region. The PROD list implies that only the specified products are to be included in the region during initialization and that all other products are to be excluded.

Note: The list of included products cannot be modified without a restart of the region. Products not nominated in the PROD list are not resident in the region while it is active.

READY={ NMREADY | *membername* }

Specifies the name of the NCL procedure, in the COMMANDS procedure library, which executes as part of system initialization after VTAM ACBs have been opened successfully.

Limits: One to eight characters

SEC={ * | NO | PARTSAF | NMSAF / *name* }

Specifies whether the region uses a security exit.

If an asterisk (*) is specified, the region uses a security exit if one has been link edited into the NM001 load module. If no security exit has been link edited, then the region uses the NMUEX01 load module if it is in an accessible load library.

If NO is specified, no security exit is used. This overrides any link edited exit or the NMUEX01 load module.

If PARTSAF is specified, a vendor-supplied partial security exit that uses SAF is used.

If NMSAF is specified, the vendor-supplied security solution is used.

If name is specified, the named load module is loaded and used as the security exit. If this load module cannot be found, then the region terminates.

Important! If an abend occurs in the exit and the requested function cannot be performed, it is regarded as a security exposure and the region terminates. Message N00303 is sent to the console as a WTO, with RC=8.

Note: For more information about security exits, see the *Security Guide*.

SSID={ NO | * | *name* }

Specifies whether to establish a connection to a SOLVE Subsystem Interface (SSI) during initialization.

If NO is specified, then no connection to a SOLVE SSI is attempted. The connection is started (or attempted) only if the SSI parameter group requests a connection.

If an asterisk (*) is specified, then an SSID of the first four characters of your product region's job name is used.

If *name* is specified, then a specific SSID must be entered.

If * or *name* is specified, then an attempt to connect to the SSI is made immediately. If it fails, it retries every *n* seconds, depending on the value in the SSI parameter group.

TZ={ *shhmmm* | GMT }

Sets the internal time zone for your product region to a specific offset or Greenwich Mean Time (GMT). *The operating system offset is ignored, and no changes will be detected or processed.* Generally, you should not specify this parameter and your region runs on system local time.

The system hardware clock is usually set to GMT. The system local time is set by the operator when the operating system is initialized. The difference between the hardware time and the local time is the time zone offset from GMT.

When the hardware time is not set to GMT, you can use the TZ parameter to set this time. The region uses the system local time as the current time but uses the specified offset when calculating GMT.

s

Is plus (+) or minus (-), depending on whether the site is ahead (+) or behind (-) GMT.

hhmm

Specifies the four-digit hours and minutes value of the time zone offset.

Limits: 15 hours

If GMT is specified, then, assuming that the hardware clock is set to GMT, the internal time is set to GMT, rather than system local time. For example, if the system is four hours ahead of GMT and the local time is 5 pm, GMT time is 1 pm. Your region will use 1 pm as the displayed 'local' time.

UDBDEFER

Specifies that all UDB open processing is to be deferred and no attempts to open UDBs are to be made before the UDBCTL command is used.

NCL automatically attempts to open DD names that start with UDB on the assumption that they are utilized as UDBs. If specific processing options are required for UDBs, such as use of LSR pools, use the UDBCTL command to open the UDB with the required options.

Selecting DD names that do not start with UDB bypasses automatic open processing and requires opening by the UDBCTL command. Typically, UDBCTL commands to open UDBs are placed in the NMINIT or NMREADY procedures.

VFSENQ={ YES | NO }

Specifies whether the VFS data set is enqueued exclusively on initialization. The default is YES. NO blocks the ENQ and allows immediate restart.

When using the ABENDCMD command to restart a started task, z/OS do not hold the new task until the old one finishes dumping. This can cause errors during startup. For example, NCL dynamic allocation requests might fail if DISP=OLD is specified. Also, VSAM data sets might fail to open if SHROPTIONS 1 or 2 are in effect.

VSAMIO={ M | S | D }

Specifies how VSAM I/O is performed in your product region:

- M specifies that all VSAM requests are performed in the main task.
- S specifies that all VSAM requests are performed in the subtask.
- D specifies that your product dynamically switches between main task and subtask, based on load.

If significant VSAM activity is anticipated, specifying S improves processing overlap on multi-CPU machines.

WTO={ YES | NO }

Specifies whether the system console is to be sent monitor class messages if no definition exists for the system console user ID (usually *pppp*OPER).

XM={ TASK / ZIIP | BEST }

Specifies whether to move some processing performed by the main task of the region from the central processor (CP) to a zIIP.

- TASK or T specifies that main task processing occurs on the CP.
- ZIIP or Z specifies that main task processing occurs on a zIIP. If a zIIP is not available, an error message is generated and processing continues on the CP.
- BEST or B specifies that main task processing occurs on a zIIP if it is available; otherwise, processing occurs on the CP.

Default: TASK or T

XOPT={ *option* | (*option, option, ...*) }

Specifies the following options (for example, for dump processing and ARM registration). This parameter can be specified more than once.

NOSXWEBU

(Default) Specifies that user security exit cannot handle web users.

SXWEBU

Specifies that user security exit has been modified to handle web users.

SDUMP

Specifies that write ABEND dumps to the SYS1.DUMP data set.

NOSDUMP

(Default) Specifies that send ABEND dumps to the normal dump data sets.

DAE

Specifies to provide DAE symptom information when writing an ABEND dump.

NODAE

(Default) Specifies that DAE symptoms information is not provided.

ARM

Specifies that the region tries to register with the Automatic Restart Manager (ARM), using an ARM element name specified through the ARMNAME= JCL parameter. If a registered region fails, the sysplex ARM restarts that region automatically.

NOARM

(Default) Specifies that the region does not register with ARM.

RLSU

Requests VSAM Record Level Sharing (RLS) for the UAMS file.

NORLSU

(Default) Specifies that VSAM RLS is not used for the UAMS file.

PVLOAD

(Default) Loads persistent variables when starting a region.

NOPVLOAD

Prevents the loading of persistent variables when starting a region.

PWMIX

Specifies that mixed case passwords are supported.

If you enable this support, consider the following important points:

- Do not share a UAMS database with a region that does not support mixed case passwords and is not using a partial security exit.
- Ensure that all regions in a multisystem environment have this support enabled.

Product Name Keys

CA SOLVE:Access has a product name key of SNAACCESS and an LMP code of ZB. The PROD product region JCL parameter specifies the name keys for the products to include in the region.

Appendix C: SYSPARMS Operands

This section contains the following topics:

SYSPARMS Summary Table

This following table provides a summary of SYSPARMS operands.

Note: After region initialization, you cannot use the SYSPARMS command to change those operands set by parameter groups.

SYSPARM	Description
ACBRETRY	Specifies whether your product region attempts to reopen the VTAM ACB.
APPLSTIV	Specifies the frequency in seconds for monitoring application status.
APPTXT1	Specifies an override description for the ACTIVE application status.
APPTXT2	Specifies an override description for the INACTIVE application status.
APPTXT3	Specifies an override description for the NOLOGONS application status.
APPTXT4	Specifies an override description for the UNKNOWN application status.
APPTXT5	Specifies an override description for the UNAVAIL application status.
APPTXT6	Specifies an override description for the NOTAPPL application status.
AUTOEXEC	Specifies whether your product region attempts to execute an unrecognized command string as an NCL procedure.
CALLSHR0	Specifies whether a subtask shares subpool zero with the main task when an &CALL statement is executed.
CDELAY	Specifies the time your product region waits before sending output to an OCS terminal when a user is entering input from the keyboard.
CONMSG	Specifies whether the message N07002 is written to the activity log each time a terminal connects to the system.
DALDEFER	Specifies whether deferred mounting is requested when allocating data sets.

SYSPARM	Description
DALRACF	Specifies whether automatic RACF protection is requested when dynamically allocating new data sets.
DALRLSE	Specifies whether data sets created by dynamic allocation are defined with the RLSE option.
DESC	Specifies the operating system description codes used for messages sent to the system console.
DYNLMAX	Specifies the maximum number of dynamic INMC links that can be concurrently active.
EDITCAPS	Specifies the default setting for the CAPS command.
EDITNULL	Specifies the default setting for the NULLS command.
EVCMDMIN	Specifies the minimum repeat frequency for the EVERY command.
HELDMSG	Specifies the default number of messages that are queued for an OCS window in HOLDING or AUTOHOLD mode, or if the window is closed.
INMCBFSZ	Specifies the INMC buffer size for INMC traffic.
INMCEX01	Defines the load module or phase name for the INMC primary security exit.
INMCEX02	Defines the load module or phase name for the INMC secondary security exit.
IPAMHB	Controls the use of heartbeats for TCP/IP INMC and APPC links.
JRNLPROC	Specifies the NCL procedure to start when a journal swap occurs.
JRNLSWAP	Indicates whether the NDB journal data set is automatically swapped if a space error occurs on the active journal.
LANG	Specifies the language code for the system.
LMSGWARN	Specifies the repeat frequency at which OCS operators are warned of lost messages.
LNKTRACE	Specifies whether a trace message is issued each time an attempt to open a session to a remote region fails.
LOCKPROC	Specifies the procedure that is invoked when a LOCK command is issued.
LOGPAGE	Specifies the number of lines the activity log has on each page.
MAIACBLN	Specifies the length of the original application name used by MAI when matching an MAI session passed between applications using the VTAM CLSDST/PASS function.
MAIACBOR	Specifies the maximum number of retries permitted when opening an MAI ACB.

SYSPARM	Description
MAIDISC	Specifies the way in which session disconnection is supported.
MAIDSSEP	Specifies a single special character that is recognized as the delimiter in MAI logon requests.
MAIENDP	Specifies the format of the session-end panel presented when an MAI-FS session ends.
MAIENDPD	Specifies whether a session-end panel is presented when an MAI-FS session ends normally.
MAIESC	Specifies how a window is terminated with MAI-FS still running.
MAIEVENT	Specifies whether MAI is to issue system EDS events.
MAIEWSAV	Specifies how MAI-FS remembers any data stream that starts with an erase-write or erase-write-alternate command received from an application.
MAIEX02	Specifies the name of an exit routine to take control whenever an MAI-OC session is started or ended.
MAIEX02S	Specifies the way MAI-OC serializes calls to the MAIEX02 exit routine.
MAIEX03	Specifies the name of an optional exit to be given control when a session jump is performed.
MAIFSTGU	Specifies the method of storage usage by MAI-FS for session management.
MAIFSTMX	Specifies the maximum number of operating system subtasks that MAI-FS uses.
MAIMDTAB	Specifies a logmode table name that contains the default logmode entries that MAI-FS uses in session requests.
MAIMENU	Specifies the name of the NCL procedure that is executed to provide the MAI menu. The default is $MAIMENU.
MAIONL	Specifies whether MAI appends a new line character (X’15’) to data sent to the target application (that is, inbound from the terminal).
MAIOPREF	Specifies a one- to five-character string, which is used as the prefix to an LU name generated by MAI-OC.
MAIOTRNS	Specifies the translate table used by MAI-OC.
MAIRESP	Specifies when MAI sends a response to the physical terminal.
MAITLOCK	Controls the use of the LOCK primary and session commands.
MAIVAUTH	Specifies whether MAI uses VTAM Authorized Path facilities during terminal I/O operations.

SYSPARM	Description
MAXRUSZ	Specifies the maximum request unit size for APPC sessions.
MENULU1	Specifies an alternate soft menu for LU1 logons.
MENUPROC	Specifies an alternate primary menu procedure name.
MODLUSER	Specifies the name of a UAMS user ID definition, present on the UAMS data set, to use as a model for dynamic user generation.
NCLEX01	Specifies the NCL general authorization exit.
NCLGBTRC	Specifies a single global variable name or a generic global variable prefix to trace as changes occur.
NCLOGTRM	Specifies whether NCL writes log message N03906 on completion of each NCL procedure.
NCLTRLFF	Specifies how many X'FF' field separators NCL places at the end of a record written to a UDB.
NCLTRMAX	Specifies the number of NCL trace messages that are generated when an NCL procedure is invoked.
NDBLOGSZ	Sets the number of VSAM logical records that are formatted as a journaling area when an NDB is created using the NDB CREATE command.
NDBOPENX	Controls whether the nominated NCLEX01 is called for &NDBOPEN.
NDBPHONX	Registers the name of the NCL phonetic exit program.
NDBRUMIN	Sets the minimum adjacent record ID (RID) range to reuse.
NDBRUSCP	Sets the percentage of used RID space to scan for reuse.
NDBSCANO	Enables or disables the scan optimizer.
NDBSUBMN	Sets the minimum number of subthreads that stay active, for any NDB, awaiting database requests that can run asynchronously.
NDBSUBMX	Sets the maximum number of subthreads permitted.
NONSWAP	(z/OS) Specifies whether this system is to run non-swappable or swappable.
NRDLIM	Specifies the number of non-roll delete messages that the system queues before deleting the oldest messages.
NSPRTINT	Sets the default timeout for solicited responses to commands sent to CA NetSpy.
OCSHLITE	Specifies the type of highlighting to use for messages appearing in OCS windows.

SYSPARM	Description
OCSTIME	Specifies whether the time appears at the end of the title line of an OCS window.
PANLBFSZ	Specifies the maximum outbound data stream size that can be generated for any terminal attached to your product region.
PANLBUFF	Specifies the maximum number of pages of virtual storage that can be used for concurrent terminal output operations.
PWEXPIRE	Specifies the number of days after which users are forced to change their password.
PWMAX	Specifies the maximum acceptable length for passwords.
PWMIN	Specifies the minimum acceptable length for passwords.
PWRETRY	Specifies the number of times an incorrect password is accepted before a logon attempt is denied.
ROUTCDE	Specifies the operating system routing codes to use for unsolicited messages sent to the system console.
SESSMSG	Specifies whether trace message N35007 is issued each time a session to a remote system opens or fails.
SMFID	Specifies the SMF record identifier to use in the generation of SMF records.
STGWRN	Specifies the number of kilobytes below the 16-MB line at which a N01801 message is issued as a WTO indicating that the storage thresholds have been exceeded.
STGWRNXA	Specifies the number of kilobytes above the 16-MB line at which a N01801 message is issued as a WTO indicating that the storage thresholds have been exceeded.
SYSCONMU	Specifies the default user ID for a master console user when it is not signed on.
SYSCONNM	(z/OS) Specifies the LU name that is assigned to system console environments.
SYSCONSO	(z/OS) Specifies whether the console user ID can default and whether signon is required.
SYSCONUI	Specifies the default system console signon name.
SYSCONXU	(z/OS) Specifies whether external console user IDs are used when signing on consoles.
SYSLOG	(z/OS)Specifies whether none, all, or unsolicited VTAM messages written to the activity log are written to the system log.

SYSPARM	Description
SYSLOGFM	(z/OS) Specifies the format of the SYSLOG lines.
TNDSREG	Specifies whether the Telnet server registers new connections with the Data Space Manager.
TRACEOPT	Specifies the trace options to apply when tracing data streams sent to or from a terminal.
USERPW	Specifies whether the NCL system variable &USERPW is available for use in MAI logon data.
VDISPLAY	Specifies how the VTAM display command (D) is processed for users with command network partitioning.
VTAMID	Specifies the system procedure name used for starting VTAM.
XABELOW	Specifies whether your product region allocates buffer storage below the 16-MB line in XA systems if all storage in the extended private area has been used.

SYSPARMS Operand Descriptions

ACBRETRY={ NO | YES }

Indicates whether your product region attempts to reopen the VTAM ACB. If YES, the system attempts to open the VTAM ACB during initialization. The system also attempts to reopen the ACB if it is closed at some time during normal processing (for example, if VTAM is shut down), at 30-second intervals.

Default: YES

APPLSTIV={ 180 | *n* }

Specifies the frequency in seconds for monitoring application status. The value can be in the range 30 through 3600. Applications are defined using SYSPARMS APPLSTAT=*applname*.

Default: 180

APPTXT1={ ACTIVE | *word* }

Specifies an override description for the ACTIVE application status.

APPTXT2={ INACTIVE | *word* }

Specifies an override description for the INACTIVE application status.

APPTXT3={ NOLOGONS | *word* }

Specifies an override description for the NOLOGONS application status.

APPTXT4={ UNKNOWN | *word* }

Specifies an override description for the UNKNOWN application status.

APPTXT5={ UNAVAIL | *word* }

Specifies an override description for the UNAVAIL application status.

APPTXT6={ NOTAPPL | *word* }

Specifies an override description for the NOTAPPL application status.

AUTOEXEC={ NO | YES }

If an unrecognized command string is entered from an OCS terminal, this operand specifies whether your product region assumes that it is the name of an NCL procedure which it then attempts to execute.

If AUTOEXEC=NO is specified, unrecognized strings are rejected with an error message.

If AUTOEXEC=YES is specified and the string is a potential NCL procedure name (that is, a valid member name), your region generates a START command for commands entered from OCS or an EXEC command for commands issued by an NCL procedure. If the string is not the name of a defined NCL procedure the command is then rejected.

Default: YES

CALLSHR0={ NO | YES }

Specifies whether subpool zero is to be shared between &CALL subtasks and your product region's main task.

When &CALL statements are executed, a subtask is attached to execute the target load module. By default, the subtask does not share subpool zero with the main task. However, if multiple called modules open and process the same VSAM data set, VSAM can suffer abends unless subpool zero is shared.

Default: NO

CDELAY=*number*

Specifies the time (in seconds) your product region waits before sending output to an OCS terminal when a user is entering input on the keyboard.

OCS terminals can receive unsolicited output at any time. If a user is entering input on the keyboard and output becomes available, a contention condition arises (on SNA terminals) which prevents a terminal from accepting output until input is complete.

This contention is broken if it continues for more than the CDELAY interval by interrupting the keyboard input and forcing the available output to the terminal. Therefore, the CDELAY parameter should be set to an interval that provides a reasonable period for uninterrupted entry of commands on the keyboard without causing excessive delay before output can be sent. The default value of 15 seconds is usually adequate. CDELAY also applies to NCL panel display.

Default: 15

Limits: 0 to 300

CONMSG={ NO | YES }

Specifies whether your product region is to write message N07002 to the activity log each time a terminal is connected to the system.

If using EASINET, this message occurs each time a terminal is returned to EASINET control after being logged on to another application. The LOGPROC NCL procedure can intercept these messages and retain statistics on terminal usage and the length of time terminals remain connected to other applications.

Default: NO

DALDEFER={ NO | YES }

Specifies whether deferred mounting is to be requested when allocating data sets.

Default: NO

DALRACF={ NO | YES }

Specifies whether automatic RACF resource protection is requested when dynamically allocating new data sets.

Default: NO

DALRLSE={ NO | YES }

Specifies whether data sets created by dynamic allocation are defined with the RLSE option to free unused secondary extents.

Default: NO

DESC={ (*n,n,n*) | NONE }

Defines the operating system descriptor codes used for messages sent to the system console, including messages associated with the *pppp*OPER user ID. Specify NONE to cancel any previously set description codes.

Note: For an explanation of the impact of specifying certain codes, see the appropriate operating system manual.

Default: NONE

Limits: 1 to 16 and NONE

DYNLMAX=*number*

Specifies the maximum number of dynamic INMC links that can be active at the same time. Enter 0 to prevent any dynamic links from being established and to disable the dynamic facility.

Default: 10

EDITCAPS={ OFF | ON }

Specifies the default setting for the CAPS command when using Panel Services. OFF specifies that entered data is to be retained in upper and lower case format. ON specifies that entered data is to be converted to upper case. The default attribute can be changed during editing by the CAPS command. When a member is saved, Edit Services retains the current CAPS setting and reinstated that value when the member is next edited regardless of the setting of the EDITCAPS operand.

Default: OFF

EDITNULL={ ON | OFF }

Specifies the default setting for the NULLS command when using Panel Services. If OFF is specified, trailing blanks are retained on each edit line. If ON is specified, trailing blanks on each edit line are converted to nulls (X'00') before being displayed for editing. Using nulls on an edit line lets you use the insert key to insert data amongst other text on the line. Use the NULLS command to change the default attribute while editing.

When a member is saved, Panel Services retains the current NULLS setting and reinstates that value when the member is next edited, regardless of the setting for the EDITNULL operand.

Default: ON

EVCMDMIN=*number*

Determines the minimum repeat period (in seconds) permitted for an EVERY command. This can be used to stop the system from being flooded with timer commands inadvertently. If a value of 0 is specified, an EVERY command is executed immediately.

Default: 10

Limits: 0 to 300

HELDMSG=(*xxx,yyy*)

xxx defines the default number of messages that are queued for an OCS window in HOLDING or AUTOHOLD mode, or where the window is closed. When this limit is reached, the earliest messages are discarded to allow the latest messages to be queued and a warning that messages are lost is sent to the terminal. *yyy* defines the maximum depth of the message queue that any user can request when using the PROFILE HOLD command. This lets you limit the size of any individual operator's message queue by overriding the default range of the PROFILE HOLD command.

Default: (200,999)

Limits: 10 to 999, and *xxx* cannot exceed *yyy*.

INMCBFSZ=*number*

Specifies the INMC buffer size (in KB) for all INMC traffic. This buffer size is the default size used for all outbound messages on all INMC sessions except for those sessions whose bind specifies a maximum RU size smaller than INMBFSZ. For these sessions the message size is equal to or smaller than the bind RU size.

Increasing INMCBFSZ can improve performance on high speed links, such as channel-to-channel or microwave links. Slow links can benefit from a lower INMCBFSZ value.

This operand can be changed at any time. However, the changed value of INMCBFSZ is not reflected in active links until they are stopped, reset, and restarted.

Default: 4

INMCEX01={ *exitname* | <u>NONE</u> }

Defines the load module for the INMC Primary Security Exit to be invoked whenever any INMC link becomes active. *exitname* is any valid module name. If you specify NONE at a later stage, the definition of any existing primary exit is canceled. This operand is applicable only for systems configured with INMC.

Default: NONE

INMCEX02={ *exitname* | <u>NONE</u> }

Defines the load module for the INMC Secondary Security Exit to be invoked whenever any INMC link becomes active. *exitname* is any valid module name. If you specify NONE at a later stage, the definition of any existing secondary exit is canceled. This operand is applicable only for systems configured with INMC.

Default: NONE

IPAMHB={ NO | (a,b) }

Controls the use of heartbeats for TCP/IP INMC and APPC links.

IPAMHB=NO disables heartbeats for INMC/APPC links using TCP/IP (regardless of the setting on the other side).

IPAMHB=(*a*,*b*) sets up heartbeats as follows:

- *a* is the heartbeat send interval, in seconds. The valid range is 10 to 100.
- *b* is the heartbeat loss toleration count. The valid range is 2 to 5.

If no heartbeats are received in (*a***b*) seconds from the other side of the session, then the session is closed. For INMC, the link is retried later.

If both sides of a link want heartbeats, then the *minimum* of each value is used; that is, the minimum of the two send intervals and the minimum of the two loss toleration counts.

Default: NO

JRNLPROC=*procname*

Specifies the NCL procedure to be started when a journal swap occurs.

Default: $NDJPROC

JRNLSWAP={ NO | YES }

Specifies whether the NDB journal data set is to be swapped automatically if a space error occurs on the active journal.

When YES is specified, the journal is swapped to the alternate journal if the file full condition occurs.

When NO is specified, the journal error prevents further updates to journaled NDBs and any updates in progress are held over until the NDB is successfully restarted.

Default: YES

LANG={ US | UK | *cc* }

Defines the language code for the system and the default language code for users.

Note: For the system recognized values that can be used to replace the *cc* option on this operand, see the *Network Control Language Reference Guide*.

Default: US

LMSGWARN=*number*

Specifies the repeat frequency at which OCS operators are warned of lost messages when in HOLDING, AUTOHOLD, or FS-HOLD mode. The warning message appears every *number* messages lost.

Default: 10

Limits: 1 to 999

LNKTRACE={ NO | YES }

Specifies whether a trace message is to be issued each time an attempt to open a session to a remote region fails.

If YES is specified, the system issues an N35006 trace message to Monitor status users each time an attempt to open a session to a remote system fails for any reason. The system retries such failures indefinitely at 60 second intervals until the session is opened or the link is stopped. If a remote system cannot be contacted, turn on this trace and examine the content of the LNKTRACE message to help resolve the cause of failure.

Default: NO

LOCKPROC=*procname*

Specifies the LOCK command procedure. This procedure is invoked when a user enters the LOCK command to lock the terminal.

Default: $NMLOCK (supplied procedure)

LOGPAGE=*number*

Defines the number of lines per page of the activity log. This operand must be executed near the start of the NMINIT initialization procedure to become immediately effective. Values of 30 to 250 can be specified.

Default: 60

MAIACBLN=*n*

Specifies the length in bytes of the original application name used by MAI when matching an MAI session passed between applications using the VTAM CLSDST/PASS function.

This operand is only applicable to shared ACBs. MAI compares the first *n* characters of the new application name with the name of the application that performed the CLSDST/PASS. A length of 0 means that the original application name must be a subset of the new application name for MAI to accept the session.

Limits: 0 through 7

MAIACBOR=*nn*

An MAI ACB name is constructed using a prefix and a numeric suffix. When an ACB fails to open, the suffix is incremented and the ACB open is retried. There are two types of failure:

- A failure because the ACB is in use or inactive (non-error situation)
- A failure for any other reason (error situation)

There is no limit set on the number of retries allowed for non-error failures.

This operand sets the maximum number of consecutive retries allowed when an MAI ACB fails to open in an error situation.

When an ACB fails to open in a non-error situation, the retry limit is reset to the value of this operand.

Default: 5

Limits: 5 to 99

MAIDISC={ YES | NO }

Specifies the way in which session disconnection is supported.

MAIDISC=YES allows use of the DISC primary and session commands to disconnect all MAI sessions and terminate the SOLVE region. You can then reconnect from the same or another terminal according to the setting of RECONN system parameter. MAIDISC=NO disables the DISC command and session commands. Disconnection can still occur due to time-out, session failure or by the DISCONN command.

MAIDSSEP=*c*

Specifies a single special character that is recognized as the delimiter in MAI logon requests.

All characters before this delimiter are interpreted as forming the logon request. All characters after the delimiter are interpreted as forming the session description. Any special character can be specified except an ampersand (&). If the separator character is the first character in the logon request, all succeeding characters are taken as both the logon request and the description.

Default: Semi-colon (;)

MAIENDP={ FULL | SHORT }

Specifies the format of the session-end panel presented when an MAI-FS session ends.

MAIENDP=FULL presents a panel containing information about the session that has ended and instructions as to how to proceed to the next session. MAIENDP=SHORT presents a panel that contains three asterisks (***) at the top left-hand corner. In both instances, additional error messages can also appear on the panel.

MAIENDPD={ YES | NO }

Specifies whether a session end panel is presented when an MAI-FS session ends normally.

If the end panel is not displayed, an automatic jump to another session or to the MAI menu is performed at session end. If a session ends because of an error condition, a session end panel is still displayed.

MAIESC={ WARN | NO | LET }

Specifies how a window is terminated with MAI-FS still running.

MAIESC=WARN displays a warning message if termination is attempted. If you then attempt to exit again immediately, the existing sessions are forcibly terminated.

MAIESC=NO prevents you to terminate a window with sessions in existence.

MAIESC=LET forcibly terminates sessions when you terminate a window, without any warning.

MAIEVENT={ YES | NO }

Controls whether MAI is to issue system EDS events.

If MAIEVENT=YES is specified, EDS events are generated when users start, stop, or change MAI sessions, or log on or off from MAI. The system EDS events are generated with a prefix of $$SYS.MAI.

MAIEVENT=YES specifies the generation of EDS events.

MAIEWSAV={ FASTJUMP | YES | NO }

Specifies how MAI-FS remembers any data stream that starts with an erase-write or erase-write-alternate command received from an application.

FASTJUMP specifies that such data streams are remembered only if the session also specifies the fast jump option. YES specifies that they are always remembered, and NO that they are never remembered. If the data stream is remembered, it can be used to subsequently refresh the screen contents.

Note: For more information, see the *Administration Guide*.

MAIEX02={ *exitname* | NO }

Specifies the name of an exit routine to take control whenever an MAI-OC session (both MAI-OC and MAI-FS) is started or ended and optionally when the VTAM ACB has been opened. The exit routine also receives notification of the LU name chosen by MAI. This routine is given information about the session to be created and can validate and alter those details, as well as having the capability of refusing the session. In addition, the routine has the capability of correlating information across session start and end calls. The MAIEX02 parameter supplies the name of the user exit routine to perform this function. MAIEX02=NO indicates that an existing exit is to be disabled or that no exit is required.

MAIEX02S={ SYSTEM | USER }

Specifies the way in which MAI serializes calls to the MAIEX02 exit routine. MAIEX02S=SYSTEM, the default, ensures that all calls to the routine are, in effect, queued within any one system, while MAIEX02S=USER only queues concurrent calls at an individual user level.

MAIEX03={ *exitname* | NO }

Specifies an exit routine to be given control when a session jump is performed.

The exit routine can send an extra data stream to the terminal. MAIEX03 parameter supplies the name of the user exit routine to perform this function. MAIEX03=NO indicates that an existing exit will be disabled or that no exit is required.

Note: For more information, see the *Administration Guide*.

MAIFSTGU={ <u>DYNAMIC</u> | STATIC }

Specifies the method of storage usage by MAI-FS for session management. This operand can be set during region initialization.

DYNAMIC is used to specify that work areas, such as VTAM RPLs, are to be allocated as needed and freed after use. This may provide greater storage reuse and a lower real storage requirement. STATIC is used to specify that all work areas are permanently allocated during session set up.

MAIFSTMX=*number*

Specifies the maximum number of operating system subtasks that MAI-FS uses. MAI-FS can utilize subtasks for two distinct purposes.

The first use is to open and close VTAM ACBs and to call any user-supplied MAI-related exits. This is called a service subtask. The second use is to conduct the VTAM session between MAI and the target application. This is called a session subtask. There can only be one service subtask, and from one to sixteen session subtasks. Each session subtask can handle any number of actual sessions, and MAI will spread the workload amongst the subtasks available. Subtasks are detached when not required.

The use of multiple subtasks can significantly improve performance on multi-processors, and can also provide greater overlap of operations on high-paging systems (if one task is paged-out, other tasks can continue).

The MAIFSTMX value can be changed at any time. If the value is reduced, MAI-FS will request termination of existing subtasks and not use them for subsequent work. Increasing the value enables additional subtasks immediately.

If the number specified is 1, MAI will use a single session subtask to manage all sessions, and will not use a service subtask. The session subtask opens and closes VTAM ACBs and calls any MAI-related exits. This can lead to reduced performance when starting or ending sessions. In addition, ACB sharing cannot be used if not running a service subtask.

If the number specified is greater than 1, one subtask will be used as a service subtask, the others as session subtasks.

Default: 5

Limits: 1 through 17

MAIMDTAB={ *name* | NO }

Specifies a logmode table name that contains the default logmode entries that MAI-FS uses in session requests.

The logmode table specified by this parameter must be assembled and linked into a load library accessible to CA SOLVE:Access.

Note: For a full description of the process by which MAI-FS selects default logmode entries, see the *Administration Guide*.

MAIMDTAB=NO specifies that an existing logmode table's use is to be disabled, or that no logmode table is required.

MAIMENU={ <u>$MAIMENU</u> | *name* }

Specifies the name of the NCL procedure that is executed to provide the MAI menu.

Default: $MAIMENU

MAIONL={ NO | <u>YES</u> }

Specifies whether or not MAI will append a new line character (X'15') to data sent to the target application (that is, inbound from the terminal).

SYSPARMS MAIONL=YES, the default, appends a new line; SYSPARMS MAIONL=NO does not.

Default: YES

MAIOPREF=*prefix*

Specifies a character string which is used as the prefix to an LU name generated by MAI-OC, for example:

```
SYSPARMS MAIOPREF=MAI02
```

If the SYSPARMS MAIOPREF operand is not coded, MAI-OC assumes a default prefix of NMMAV.

Limits: One to five characters

MAIOTRNS=(*xx,yy*)

When data is received from an application across an MAI-OC session, MAI-OC translates, by default, any characters below X'40' to an underscore (_) before displaying it at an OCS window and leaves all other data intact. The MAIOTRNS operand can be used to alter the translate table used by MAI-OC in this process. The value *xx* is two hexadecimal digits representing a value to be translated and *yy* two hexadecimal digits representing a value into which *xx* is to be translated. The *yy* value must represent a *printable* character.

For example, to translate a hexadecimal zero (X'00') to a blank, specify:

```
MAIOTRNS=(00,40)
```

MAIRESP={ <u>AFTER</u> | BEFORE }

Specifies whether MAI sends a response to the physical terminal.

When an application sends data to MAI, it usually requests a response. If running with MAIRESP=AFTER, MAI will not send this response until the data has been sent to the physical terminal, thereby ensuring integrity.

This can be a considerable time later if, for instance, the terminal user is viewing another session. If MAIRESP=BEFORE is specified, the response is sent before the data is sent to the physical terminal. However, if the application sends more data before the responded-to data is sent to the terminal, the response to subsequent data is withheld. Thus only one message is ever responded to ahead of time.

The use of MAIRESP=BEFORE can speed up operations in circumstances where an application sends unsolicited messages, one after another, without waiting for input from the terminal. It is also useful in circumstances where an application will not clean up resources until a last message has been acknowledged. Consider the case of a TSO session being cancelled. TSO writes a message to the terminal informing the user that the session has been cancelled, but until that message is responded to, it will not terminate the address space. If the user is viewing another session at the time, MAI will normally not respond to this message. However, with MAIRESP=BEFORE, the session will be terminated successfully, and the user will see the last message on return to the TSO session.

There is some loss of integrity using MAIRESP=BEFORE, but it will not affect most applications. Only those applications which take some important action based upon the assumption that the data has been displayed on the physical terminal are affected.

Use of MAIRESP=BEFORE also prevents MAI-FS from mirroring negative responses to the application. Normally, if MAI-FS receives an unrecoverable negative response from the terminal, that response will be sent on to the application for it to perform its own recovery action. However, with MAIRESP=BEFORE, MAI positively responds to the application in advance, and receipt of a negative response from the terminal results in the cancellation of the session.

The MAIRESP operand of the DEFLOGON command may be used to modify this value for particular applications.

MAITLOCK={ YES | NO }

Controls the use of the LOCK primary and session commands.

MAITLOCK=YES allows use of the LOCK commands.

These commands cause the display of a panel requesting the user’s password. When this is entered, the user can resume work at the terminal. MAITLOCK=NO disables use of the commands. This operand controls whether the LOCK session command is made available. The time-out action of LOCK always locks the terminal.

MAIVAUTH={ YES | NO }

Specifies whether MAI will use VTAM Authorized Path facilities during terminal I/O operations.

Specify NO if authorized path facilities are not to be used for MAI sessions.

MAXRUSZ=*number*

Specifies the maximum request unit size for APPC sessions. If necessary this value overrides the LU6.2 session BIND parameters. The value specified on this command applies to new sessions established after the command is complete. The default value is 4096. The value specified is converted to a single byte RU size of the form X'*ab*' where number=*a**2***b*. Values must be in the range 128 to 32768 and the values which cannot be converted directly are rounded down.

MENULU1=*procname*

Specifies an alternate menu for LU1 logons.

Default: $NMPMLU1 (supplied procedure)

MENUPROC=*procname*

Specifies an alternate primary menu procedure name.

Default: $NMPMENU (supplied primary menu procedure)

MODLUSER={ *userid* | <u>NONE</u> }

Specifies the name of a UAMS user ID definition, present on the UAMS data set, to be used as the model for dynamic user ID generation. Use the following scenarios to determine the most appropriate action for your system:

- If you do not want to specify a model for dynamic user ID, specify NONE or let it default.
- If this operand is coded and a user attempts to log on to a product region with an unknown user ID, but their password is that associated with the model user ID, the system automatically creates a new user ID for the unknown user. This ID has exactly the same attributes and privileges as the model user ID.
- If a password security exit is implemented, the exit can alter the model name that is to be used.
- If a full security exit is implemented this operand is ignored.

Default: NONE

NCLEX01={ *exitname* | NO }

Specifies the name of the NCL authorization exit. This exit provides security for NCL procedures and &SMFWRITE verbs.

Note: For full information about NCLEX01, see the *Security Guide*.

If NCLEX01 is frequently invoked, use the LOAD MOD=*exit_name* command to load a re-entrant version of the exit into the region to eliminate the overhead of loading the exit for each authorization.

If NO is specified, then the current exit name is deleted and exit invocation is stopped.

Note: The NCLEX01 load module is executed under a subtask and can therefore issue I/O and WAIT operations without impacting the main system.

NCLGBTRC= { *name* | *prefix }**

Specifies a single global variable name, or a generic global variable prefix that is to be traced as changes occur to them. Each time an assignment occurs into a traced global variable, a log record is written that identifies the process performing the assignment. The first eight bytes of data are also traced.

If *name* is specified, only one global variable is traced. If *prefix* is specified, all global variables that start with the nominated prefix, excluding the standard global variable prefix, are traced as their values change.

The trace record is written as a single message, N23312, to the activity log. Tracing is turned off if no name or prefix is specified.

NCLOGTRM={ NO | YES }

Specifies whether NCL writes a log message on completion of each NCL procedure. If YES is specified, the log message provides statistics on NCL processing units used by the procedure.

Default: NO

NCLTRLFF={ ONE | MULT }

Specifies how many X'FF' field separators NCL places at the end of a record written to a UDB. If MULT is specified, and a record is written to a UDB where the record contains multiple null variables at the end, one X'FF' field separator is appended to the UDB record for each 'null' variable on the &FILEPUT or &FILEADD statement.

If NCL-format UDBs are created for processing by external systems, the use of this operand should be consistent.

Default: ONE

NCLTRMAX=*number*

Specifies the maximum number of NCL trace messages that are generated in any one invocation of an NCL procedure. If 0 is specified, all tracing is inhibited.

Default: 100

Limits: 0 to 9999

NDBLOGSZ=*n*

Sets the number of VSAM logical records that will be formatted as a journaling area when an NDB is created using the NDB CREATE command. This journal area provides transaction integrity across system failures.
If the LOGSIZE parameter is specified on NDB CREATE, it overrides this value.

This value can be changed prior to issuing an NDB CREATE command, to change the journal size for that database.

Journal size is influenced by the possible complexity of an add, update, or delete operation on the database, which in turn depends on such things as the size of the data record, and the number of keys being added. The journal automatically extends if it is under-allocated.

Note: For more information, see the *Network Control Language Programming Guide*.

Default: 40

Limits: 10 to 200

NDBOPENX={ NO| YES }

Controls whether the nominated NCLEX01 is called for &NDBOPEN.

NDBPHONX=*name*

Registers the name of the NCL phonetic exit program.

NDBRUMIN=*nnn*

Sets the minimum number of consecutive RID numbers that can be reused. You can change this operand at any time.

Default: 20

Limits: 10 to 100

NDBRUSCP=*nnn*

Sets the percentage of RID space to be scanned to collect RIDs for reuse. Values of 95 or greater cause the complete NDB to be scanned. For an NDB that is scanned daily, a default value of 15 means that the NDB is scanned completely every week. You can change this operand at any time.

Default: 15

Limits: 5 to 100

NDBSCANO= { NO | <u>YES</u> }

Enables (YES) or disables (NO) the scan optimizer.

Notes:

- The setting of the NDBSCANO value does not affect NDBs that are already started at the time the command was issued.
- An individual NDB can override the setting of the NDBSCANO command using the OPTIMIZE operand of the NDB START command.

Default: YES

NDBSUBMN=*n*

Sets the minimum number of subthreads that will stay active, for any NDB, awaiting database requests that can run asynchronously (&NDBSCAN and &NDBGET requests). When a database request arrives that can be run asynchronously, the database handler starts a separate copy of itself to run that request, unless there are already NDBSUBMX subthreads running. As the subthreads run out of work, they terminate unless the NDBSUBMN limit is reached.

Default: 3

Limits: 0 to 20

NDBSUBMX=*n*

Sets the maximum number of subthreads allowed. See NDBSUBMN.

Default: 5

Limits: 1 to 20

NONSWAP={ YES | NO }

Specifies whether this system is to run non-swappable (YES), or swappable (NO). This operand is valid only if your product region is running authorized.

For z/OS and MSP, your product region makes itself non-swappable automatically before system initialization. The NONSWAP operand can be utilized to change this status either during initialization or at any time after.

For VOS3, if your product region is authorized, it runs swappable, by default. It can be changed to run non-swappable by specifying YES. However, once running non-swappable, it is not possible to change the status back to swappable without stopping and restarting the system.

Note: For z/OS and MSP, optional features which require non-swappable operation blocks attempt to revert to swappable operation.

NRDLIM=*number*

Specifies the maximum number of NRD messages that the system queues at any time before discarding the oldest messages. This queue is used by the NRDRET command to refresh the OCS NRD message display.

Default: 200

Limits: 10 to 10000

NSPRTINT=*interval*

Sets the interval between retries when trying to connect to a CA NetSpy region. It is an integer number of seconds.

Default: 30

Limits: 10 to 600

OCSHLITE={ NONE | REVERSE | BLINK | USCORE }

Specifies the type of highlighting to be used for messages in OCS windows normally displayed in high intensity. The following options can be specified:

NONE specifies that messages are presented without change.

REVERSE specifies that messages are presented in reverse video.

BLINK specifies that messages are to blink.

USCORE specifies that messages are underscored.

For terminals that support color, the default color applies. This operand is ignored for terminals that do not support IBM extended highlighting and does not apply to individual high intensity fields resulting from comment lines from NCL procedures that commence with a plus sign (+) and that use the at symbol (@) field highlighter.

Default: NONE

OCSTIME={ NO | YES }

Specifies whether the time appears at the end of the title line of an OCS window. If YES is specified, the current time in the format HH.MM is placed at the left hand end of the title line of an OCS window each time the window display is updated. This allows operators to determine when the last message occurred if the terminal has been left temporarily unattended. Specifying NO resets this option.

Default: YES

PANLBFSZ=*number*

Specifies the maximum outbound data stream size (in KB) that can be generated for any terminal attached to your product region.

If your network contains terminals that might receive large data streams (for example, complex extended data stream screen formats) you might need to increase the PANLBFSZ operand. A message is issued if an attempt is made to display a panel that is too large for the current PANLBFSZ setting.

Default: 16

Limits: 4 to 20

PANLBUFF=*number*

Specifies the maximum number of pages of virtual storage to be used for concurrent terminal output operations. If 0 is specified, no limit is imposed.

This parameter acts as a throttle for simultaneous output to large numbers of terminals (for example, during broadcast processing or when starting a large EASINET network). Increase this number to speed up network start-up and broadcasting, decrease it to throttle back these activities.

Before increasing this value, consider any effects an increase in virtual storage usage might have on other functions.

Default: 40

Limits: 0 to 32767

PWEXPIRE=*number*

Specifies the number of days after which users are forced to change their password. This operand takes effect the next time a user logs on. This operand has no effect if a security exit is in force that replaces the UAMS password maintenance functions. Enter a value of 0 to disable automatic password expiry.

Default: 30

Limits: 1 to 366

PWMAX=*number*

Specifies the maximum acceptable length for passwords. This operand only takes effect from the next password change a user makes. The operand has no effect if a security exit is in force that replaces the UAMS password maintenance functions.

Default: 8

Limits: Value of PWMIN to 8

PWMIN=*number*

Specifies the minimum acceptable length for passwords. This operand only takes effect from the next password change a user makes. The operand has no effect if a security exit is in force that replaces the UAMS password maintenance functions.

Default: 3

Limits: 1 to the value of PWMAX

PWRETRY=*number*

Specifies the number of password violations that are accepted before a logon attempt is denied. If the number is reached, a warning message is sent to all terminals with Monitor status, advising them of the user ID and terminal involved in the violation. This operand has no effect if a security exit is in force that replaces the UAMS password maintenance functions.

Default: 2

Limits: 1 to 10

ROUTCDE=(*n,n,n*)

Specifies the operating system routing codes to use for unsolicited messages sent to the system console (that is, to the *pppp*OPER user ID).

Note: For an explanation of the impact of specifying certain codes, see the appropriate operating system guide.

Default: 1,8,11

Limits: 1 to 16

SESSMSG={ NO | YES }

Specifies whether trace message N35007 is issued each time a session to a remote system opens or fails. If YES is specified, the system issues the trace message to Monitor status users, each time a session to a remote system is opened or fails. The system retries any failures indefinitely at intervals specified in the LINK command until the session is opened or the link is stopped.

This option can be particularly useful in an INMC operation, as it might be the only way to identify the fact that not all sessions making up a link are operational. If any sessions are open the link remains operational-the failure of an individual session does not disrupt traffic, but can affect performance. The default is NO.

Default: NO

SMFTRACE=*number*

Requests the dump of written SMF records.

STGWRN=*number*

Specifies the number of kilobytes below the 16-MB line at which an N01801 message is issued as a WTO indicating that the storage thresholds have been exceeded. This message can be repeated at 30-second intervals until the storage use drops below the threshold.

Default: 0 (no warning limit)

Limits: 0 to 16,000

STGWRNXA=*number*

Specifies the number of kilobytes above the 16-MB line at which an N01801 message is issued as a WTO indicating that the storage thresholds have been exceeded. This message can be repeated at 30-second intervals until the storage use drops below the threshold.

Default: 0 (no warning limit)

Limits: 0 to 1,000,000

SYSCONMU=*name*

Specifies the default user ID for a master console that issues commands to your region when it is not signed on to the security system in use.

SYSCONMU can be set only during NMINIT. For master consoles that are not signed on to your security system, this user ID always applies, regardless of the setting of the SYSPARM SYSCONSO.

If this default user ID is not defined to your security system, then the user ID .MASTOP (which has a hard-coded profile) is used.

Note: For more information about master console user ID requirements, see the *Security Guide*.

Default: *pppp*MSOP, where *pppp* is the system user prefix

Limits: One to eight characters

SYSCONNM={ EXTMCS | ALL }

Specifies the LU name that is assigned to system console environments. The following options can be specified:

- EXTMCS specifies consoles with console IDs in the range 0 to 99. The consoles are named CONSOLE (0) or CONS#*nn*, where *nn* is 01 to 99. Console IDs outside this range (that is, extended MCS consoles) use the extended MCS console name.
- ALL specifies that the LU name is always the z/OS-assigned console name. This means that Console 0 is INTERNAL, the master is MASTER, and so on.

Default: EXTMCS

SYSCONSO={ DEFAULT | NO | REQUIRED }

Specifies whether the console user ID can default and whether signon is required. It is relevant to all operating system environments. The following options can be specified:

- DEFAULT specifies that the default SYSCONUI user ID name is used to sign on if the system console user ID is not defined in the UAMS security system.
- NO specifies that all consoles are signed on using the SYSCONUI user ID name.
- REQUIRED specifies that the user ID must be defined, otherwise the signon fails. This option is typically only used if SYSCONXU=YES is in effect.

Note: There are interactions between the SYSCONSO and SYSCONXU parameters.

Default: DEFAULT

SYSCONUI=*name*

Specifies the default system console signon name. The *name* must be a valid user ID.

Note: The user ID is not used unless it exists in the security system in use.

Note: For more information about console user ID requirements, see the *Security Guide*.

Limits: One to eight characters

SYSCONXU={ NO | YES }

Specifies whether or not external console user IDs are to be used when signing on consoles. The following options can be specified:

- NO specifies that the system-supplied user ID is not used.
- YES specifies that the system-supplied user ID is used. However, it is only used if the system passes a RACF UTOKEN with the command. If no token is passed, or the console is not signed on to RACF, the console is treated as not signed on, and the action taken depends on the value of SYSCONSO. If the master console user ID, *BYPASS*, is seen, a special internal user ID of .MASTOP is used to indicate that this is the master console. It is not signed on to RACF.

Note: There are interactions between the SYSCONXU and SYSCONSO parameters.

Default: NO

SYSLOG={ NO | YES | PPO }

Specifies whether none, all, or unsolicited VTAM messages written to the activity log, are also written to the system log. The following options can be specified:

- NO writes no messages to the system log.
- YES copies everything written to the activity log to the system log.
- PPO writes all unsolicited VTAM messages to the system log and the activity log.

Default: NO

SYSLOGFM={ MVS | MSP }

Specifies the format of the SYSLOG lines produced if SYSLOG=YES or SYSLOG=PPO.

Format a line with four zeroes for ROUTCDE, time, and user ID in the JOBID column.

TNDSREG={ NO | YES }

Specifies whether the Telnet Server will register new connections with the Packet Analyzer, for use by any CA NetMaster NM for TCP/IP regions on the same z/OS image.

Setting NO means connections are not registered.

Default: YES

TRACEOPT=*cc*

Specifies the trace options to be applied when tracing data streams sent to or from a terminal. The value of *cc* is the character representation of a hexadecimal byte.

The following bit values represent valid trace options:

`B'10000000'(X'80')`	Trace only first 256 bytes of each message.
`B'00000001'(X'01')`	Trace output before compression.
`B'00000010'(X'02')`	Trace output after compression.
`B'00000100'(X'04')`	Trace input from terminal.

Default: 06

Example:

```
SYSPARMS TRACEOPT=80
```

indicates that the tracing options required correspond to a hex byte with a value of X'80'. This byte in turn represents an 8-bit string with the value:

```
B'10000000'
```

Example:

The value specified on the TRACEOPT parameter can be any combination of the four options, expressed as a hex character, for example:

```
SYSPARMS TRACEOPT=84
```

indicates a request to trace the first 256 bytes received from the terminal, with the data stream being written to the activity log. Data recorded can then be examined using the standard online log browse facilities.

Tracing is started and stopped by the LUTRACE command.

USERPW={ NO | YES | VERIFY }

Specifies whether the NCL system variable &USERPW is available for use in MAI logon data. &USERPW represents the user's product region password and is used when MAI sessions are created. This operand allows installations to control the availability of the &USERPW variable to MAI.

If YES or VERIFY is specified, the password is encrypted in storage so that it is not available in plain text. Specifying VERIFY indicates that MAI should prompt users for their password when a session is updated or added that contains &USERPW in the logon data.

Default: VERIFY

VDISPLAY={ CMD | MSG | ANY }

Specifies how the VTAM display command (D) is processed for users with command Network Partitioning. The following options can be specified:

- CMD lets users display only those resources within their defined command partitions.
- MSG lets users display only those resources within their defined message partitions. This can be used where resources within command and message partitions differ. In this case, the Inactive or Active status of message tables is ignored and all message tables are searched.
- ANY lets users display any resource, regardless of the resources specified within their partitions.

Default: CMD

VTAMID=*procname*

Specifies the system procedure name used for starting VTAM. If any other procedure name is used, then your product region must be informed of it so that the correct name can be used when generating VTAM commands.

Default: (ACF/VTAM and VTAM-G) NET or (ECS/VTAM) VTM

Limits: One to eight characters

XABELOW={ NO | YES }

Specifies whether your product region is to allocate buffer storage below the 16-MB line in XA systems if all storage in the extended private area has been used. NO means that if all XA storage in the product region address space has been used, further requests for XA storage by other processes fails, even if non-XA storage is still available below the line. Do not change this unless your installation requires your product region to run with a severely limited extended private area.

Default: NO

Appendix D: Data Set Descriptions

This section contains the following topics:

Data Set Types

In the following tables, the Types column contains information about the types of data sets.

The following table explains the data set types:

Types	Explanation
Run-time PDS	Shared non-VSAM SMP target files that are allocated during installation, and used by an active region, for example, *dsnpref*.SM50.*ccdsname*.
Run-time PDSE	Shared non-VSAM SMP target files that are allocated during installation and used by an active region, for example, *dsnpref*.SM50.*ccdsname*. Program objects that must be executed from a PDSE are stored in these files.
Other PDS	Shared non-VSAM PDS created separately to this product.
Run-time PDS (external)	Shared non-VSAM SMP target libraries that are not used by the region.
Shared run-time VSAM (MODSDIS / NETINFO / NSCNTL / PANLDIS / OSCNTL / UAMS)	Files that are identified by the unique data set prefixes you enter when you set up your regions. You can choose to select a data set prefix for each file. You can make the data set prefix the same, or different from those you have used for other files. These run-time files are used by an active region. These files will be shared by more than one region. VSAM data sets that can be shared in this manner are defined using SHAREOPTIONS(3,3).
Distribution	SMP distribution libraries that are used during the installation and maintenance of a product, for example, *dsnpref*.SM50.*ccdsname*.

Types	Explanation
Local Sequential	Run-time sequential files that are allocated during setup to be used by an active region, and cannot be shared between multiple regions, for example, *dsnpref.rname.dsname.*
Staging	Shared non-VSAM files that are used during the installation process to store sequential copies of files including the product panels, MODS, and OSCNTL files, for example, *dsnpref*.SM50.*ccdsname*.S. These sequential copies are then merged to create the run-time files. If you intend to create additional regions at a later date, you can use the staging files as input to the setup process.
Local PDS	Run-time PDS files that are allocated during setup to be used by an active region, and cannot be shared between multiple regions, for example, *dsnpref.rname.dsname*.
Local VSAM	Run-time VSAM files that are allocated during setup to be used by an active region, and cannot be shared between multiple regions, for example, *dsnpref.rname.dsname*.
Shared PDS	Run-time PDS files that are allocated during setup to be used by an active region, and can be shared between multiple regions, for example, *dsnpref*.SM50.*dsname*.
TESTEXEC	File that is identified by the unique data set prefix you enter when you set up your region.

Installation and Setup

The Install Utility uses the following data sets:

***dsnpref*.SM50.INSTDB**

Stores your site-specific installation values, which can be reused in future installations of products in the same suite. The installation software allocates this data set the first time you perform an installation. The setup and maintenance software also uses this data set.

***dsnpref*.SM50.INSTALL.JCL**

Contains the installation JCL members generated after the installation software has collected your site-specific installation values.

Before you generate the JCL, specify the library where you want to store the generated JCL. INSTALL.JCL is the default JCL library.

Use an empty data set each time you perform the installation. An empty data set helps ensure that the jobs in your JCL library are the only ones required for the current installation.

***dsnpref*.SM50.*rname*.JCL**

***dsnpref*.SM50.*ssiname*.JCL**

Contain the setup JCL members generated after the setup software has collected the setup values for your region or SOLVE SSI.

These data sets contain the setup jobs, including the members that contain the execution JCL required to run the product components.

***dsnpref*.SM50.VTAM.JCL**

Contains the JCL members generated after the Create VTAM Definitions software has collected your values.

This data set contains the generated VTAM major node and JCL to assemble mode tables.

***dsnpref*.SM50.FIX.SP*n*.JCL**

***dsnpref*.SM50.FIX.DASD.JCL**

Contains the maintenance JCL members after the maintenance software has collected your maintenance values.

This data set contains the maintenance jobs that you run to apply maintenance to your installed products.

Product Components

Your product operates with a combination of common and product-specific components.

The installation process and these components have data set requirements.

Management Services Data Sets

AC2DEXEC

Is the SMP DLIB that contains the same distributed NCL procedures as the CC2DEXEC run-time library.

Type: Distribution

AC2DJCL

Is the SMP DLIB that contains the same members as the CAIJCL installation data set.

Type: Distribution

AC2DMAC

Is the SMP DLIB that contains the same information as the CC2DMAC library.

Type: Distribution

AC2DMOD

Is the SMP DLIB that contains the modules used by SMP to build the load modules in the run-time LOAD libraries.

Type: Distribution

AC2DSAMP

Is the SMP DLIB that contains the same information as the CC2DSAMP library.

Type: Distribution

AC2DVSMI

Is the SMP DLIB that contains the same information as the CC2DVSMI library.

Type: Distribution

CAIJCL

Is the PDS that contains the installation and maintenance JCL members.

Type: Run-time PDS

CAILINK

Is a PDS run-time load library that contains various NetView exits. Include in the STEPLIB DD concatenation for NetView if you are using the NVC component.

If your auditors want CA Auditor Product Descriptor Module (PDM) to know of your product running on your system, include this library in the system linklist.

Type: Run-time PDS

CAILPA

Is a separate PDS run-time load library containing modules that must be executed from the LPA. The data set contains only CNMNETM plus its aliases.

Type: Run-time PDS

CC2DEXEC

Is the PDS that contains distributed NCL procedures and is concatenated after the TESTEXEC data set in DD COMMANDS. Collectively these data sets make up the procedure library.

Members in CC2DEXEC must not be changed.

Important! If modifications are required, we recommend that you create an SMP/E ++USERMOD to record and control the changes, and then copy the member to TESTEXEC. Alternatively, you can copy the distributed member to the region’s TESTEXEC data set for modification.

Type: Run-time PDS (external)

Run-time DDName: COMMANDS

CC2DLMD0

Is the PDS into which the TSO interface load modules are linked during installation.

Type: Run-time PDS (external)

Run-time DDName: STEPLIB (TSO)

CC2DLOAD

Is the load library. This load library must be APF-authorized. This means that the run-time load library, as referenced by the STEPLIB DD statement, must be defined in the operating system APF list. If no STEPLIB DD statement is used, the program load modules must reside in one of the existing authorized linklist libraries. Ensure that these requirements are met before attempting to start your region.

You can install your product into its own installation load library and copy the load modules across to the system load library.

Type: Run-time PDS

Run-time DDName: STEPLIB

CC2DMAC

Is the PDS that contains macros and copybooks for the sample assembler programs, which are also distributed in source form in the AC2DMAC library.

Type: Run-time PDS (external)

CC2DSAMP

Is the PDS that contains various sample exits, utilities, source code, and JCL. The members contain documentation.

Type: Run-time PDS (external)

CC2DVSMI

Is the interim staging data set to which MODS, OSCNTL, PANEL, NETINFO, ICOPANL, and RAMDB files are unloaded. They are later copied to the run-time MODSDIS, OSCNTL, PANLDIS, NETINFO, ICOPANL, and RAMDB files.

Type: Staging

FMTDUMP

Is the data set the region uses for a formatted dump.

Type: SYSOUT

Run-time DDName: FMTDUMP

LOG

Is the data set the region uses for the hardcopy log.

Type: SYSOUT

Run-time DDNames: LOG1 through LOG9

MODSDIS

Is the VSAM data set that contains the MODS database. The region uses the MODS database for control information for processing that includes standard message information, online help text, menus, Print Services Manager (PSM) definitions, and Report Writer definitions.

Type: Shared run-time VSAM

Run-time DDName: MODSDIS

Note: For information about the allocation and usage of MODS databases, see the *Managed Object Development Services Programmer and Administrator Guide*.

MODSUSR

Is a data set that has the same functions as MODSDIS and is intended for your own records. You can have separate records for test and production environments.

Type: Local VSAM

Run-time DDName: MODSUSR

NETINFO

Is the VSAM data set that contains various error codes (for example, 3274 error codes and SNA sense codes).

Type: Shared run-time VSAM

Run-time DDName: NETINFO

NML0G01 through NMLOG03

Are the VSAM data sets to which the region's LOGPROC NCL procedure ($LOPROC) writes all activity log messages for subsequent online browsing from a terminal.

Type: Local VSAM

Run-time DDNames: NMLOG01 through NMLOG03

OSCNTL

Is the VSAM data set used by the region to store MDO definitions and compiled object class specifications.

Type: Shared run-time VSAM

Run-time DDName: OSCNTL

PANLDIS

Is the VSAM data set that stores full-screen panels defined by using the online editor. These panels are used by NCL procedures and Panel Services.

Your product provides the facility for multiple panel data sets per region, allowing the data sets to be concatenated and different data sets to be available to different users.

Type: Shared run-time VSAM

Run-time DDName: PANLDIS

PANLUSR

Is a data set that has the same functions as PANLDIS and is intended for your own records. You can have separate records for test and production environments.

Type: Local VSAM

Run-time DDName: PANLUSR

PARMLIB

Is the data set (*dsnpref*.SM50.PARMLIB) that contains the setup parameters for the product components in a region.

Type: Shared PDS

Run-time DDNames: DSIPARM and SXCTL

PSPOOL

Is the data set used to store printed output handled by the PSM facility.

Type: Local VSAM

Run-time DDName: PSPOOL

TESTEXEC

Is the data set (*dsnpref*.*rname*.TESTEXEC) that is concatenated as the first of the data sets forming the COMMANDS DD. The data set is used for the following procedures:

- User-written NCL procedures
- Modified versions of supplied procedures that have been copied from the distributed libraries
- Installation modified INIT and READY procedures

The setup process allocates TESTEXEC as a local data set.

Type: TESTEXEC

Run-time DDName: COMMANDS

UAMS

Is the User ID Access Maintenance Subsystem (UAMS) data set containing the security definitions for users authorized to use the region.

In a shared DASD environment where multiple regions are operative, perhaps connected by Inter-Network Management Connection (INMC), you can define one UAMS data set that all regions can share. The regions use reserve and release logic during accesses, and ensure the integrity of the data set in a shared DASD environment. If only one data set is used, operators using the Remote Operator Facility (ROF) to connect to another region have identical authority and privileges in both regions. This situation may not be satisfactory if one region is dedicated to testing and the other to production. In this case, use two UAMS data sets, allowing a user to be profiled differently in the two regions.

If the installation uses a security exit to replace the UAMS component entirely, the UAMS data set is not required.

When logging on or changing UAMS records, a RESERVE is issued for NMUAMS. The RESERVE has the following attributes:

- Qname: NMUAMS (padded to eight characters with blanks)
- Rname: 44-character blank padded UAMS data set name
- Scope: SYSTEMS
- Level: EXCLUSIVE
- UCB: UAMS UCM address

If the sysplex is in a STAR configuration, you can convert this RESERVE to a GRS global ENQ. If the sysplex is a RING configuration, do not convert the RESERVE.

Type: Shared run-time VSAM

Run-time DDName: USERIDS

VFS

Is the data set used by the region for internal processing activities and as a general database for use by the various products. The VFS is a common database used to store many record types and must not be shared across regions.

Type: Local VSAM

Run-time DDName: VFS

SOLVE Subsystem Interface Data Sets

SSIDB

Is the data set that acts as a disk-based backup to the nonvolatile part of the SOLVE SSI database. The data set contains saved application data and allows data to be preserved across restarts.

Type: Local VSAM

Run-time DDname: SSIDB

SSIPARM

Is the data set that contains the setup members for the SOLVE SSI.

Type: Run-time PDS

Run-time DDname: SSIIN

Access Services Data Sets

AC16EXEC

Is the SMP DLIB that contains the distributed NCL procedures as in the CC16EXEC run-time library.

Type: Distribution

AC16SAMP

Is the SMP DLIB that contains the same information as the CC16SAMP library.

Type: Distribution

AC16VSMI

Is the SMP DLIB that contains the same information as the CC16VSMI library.

Type: Distribution

AC16XML

Is the SMP DLIB that contains the same XML data as the CC16XML library.

Type: Distribution

ACDB

Is the database that stores MAI session definitions, global session search criteria, and when the SOLVE:Access region is a member of a VTAM generic resource group, disconnect and single sign-on user tracking information. The VSAM options for this file are specified in the ACDB parameter group and when running in a VTAM generic resource environment, must be configured for VSAM RLS.

Type: Shared Run-time VSAM

Run-time DDName: ACDB

CC16EXEC

Is the PDS that contains distributed NCL procedures and is concatenated after the TESTEXEC data set in DD COMMANDS. Collectively these data sets make up the procedure library.

Members must not be changed.

Important! If modifications are required, we recommend that you create an SMP/E ++USERMOD to record and control the changes, and then copy the member to TESTEXEC. Alternatively, you can copy the distributed member to the region's TESTEXEC data set for modification.

Type: Run-time PDS

Run-time DDName: COMMANDS

CC16SAMP

Is the PDS that contains various sample exits, utilities, source code, and JCL. The members contain documentation.

Type: Run-time PDS (external)

CC16VSMI

Is the interim staging data set to which MODS, OSCNTL, and PANEL files are unloaded. They are later copied to the run-time MODSDIS, OSCNTL, and PANLDIS files.

Type: Staging

CC16XML

Is the PDS that contains the XML data used by CA MSM.

Type: Run-time PDS (external)

EASIUDB

Is the data set used by EASINET for EASIUSER definitions.

Type: Local VSAM

Run-time DDName: EASIUDB

TRFILE

Is the data set used by MAI to contain Session Replay Facility trace details.

Type: Local VSAM

Run-time DDName: TRFILE

Index